User manual

DA14580/581/583 Bluetooth Smart development kit – Basic

UM-B-025

Abstract

This document describes the Bluetooth Smart Development kit - Basic based on DA14580/581/583. It helps customers to set up the hardware development environment, install required software and quickly start product development based with help of example source code.

Contents

Abstract **1**

Contents **2**

Figures **3**

Tables **3**

1 Terms and definitions **4**

2 References **4**

3 Introduction **5**

3.1 Hardware content 5

3.2 Web content 6

3.2.1 Software Development Kit content 6

3.2.1.1 Tools 6

3.2.1.2 SDK documents 6

3.2.1.3 SDK source code examples (created with Keil IDE) 7

3.3 DA14580 – DA14581 basic kit 9

3.3.1 Differences between the DA14580/580/583 9

3.4 Pinning 11

3.4.1 PCB design and functionalities 12

3.4.2 Schematics and layouts 14

3.4.2.1 DA1580DEVKT-B_va layout 14

3.4.2.2 DA1581DEVKT-B_va layout 16

3.4.2.3 DA1583DEVKT-B_va layout 18

3.4.3 Configuring the DA14580/581 basic kit-board by jumper settings 20

3.4.4 Configuring the DA14583 basic kit-board by jumper settings 22

4 Installation of tools and drivers **23**

4.1 Keil 23

4.2 SEGGER Jlink driver 24

4.3 FTDI driver 24

4.4 Tera Term 25

5 Quick start: Downloading software **26**

5.1 Software downloaded via Keil IDE: USB (via SWD) to SRAM 27

6 Using the demo kit **29**

6.1 Run an example on the DA14580/581 /583 29

7 Power Management: measuring current **34**

Appendix A Layout DA14580/581 ..... **36**

Appendix B Connections of J10 and J11 ..... **36**

Appendix C Layout DA14583 ..... **37**

Appendix D Connections of J10 ..... **37**

Appendix E Using the smart snippets CLI **38**

Appendix F Issues with opening a project for the first time **39**

F.1 Issue description: 39

F.2 Possible causes: 39

F.3 Under which circumstances user will encounter this error: 39

© 2015 Dialog Semiconductor

F.4 Proposed solution: 39

Appendix G 'Blinky' Test software 40

Appendix H Latency Timer of FTDI cable 42

8 Web-Links 43

9 Revision history 43

Contacting Dialog Semiconductor 44

Figures

Figure 1: DEVKT – Basic Kit 5
Figure 2: QFN40 pin assignment from datasheet 11
Figure 3: Components on PCB top layer of the DA14580/581Basic kit 12
Figure 4: Topview of PCB DA14580/581 with components and functionalities 12
Figure 5: Components on PCB top layer of the DA14583 Basic kit 13
Figure 6: Topview of PCB DA14583 with components and functionalities 13
Figure 7: DA14580DEVKT-B_vb part 1 14
Figure 8: DA14580DEVKT-B_vb part 2 15
Figure 9: DA1581DEVKT-B_va part 1 16
Figure 10: DA1581DEVKT-B_va part 2 17
Figure 11: DA1583DEVKT-B_va part 1 18
Figure 12: DA1583DEVKT-B_va part 2 19
Figure 13: DA14580/581 (Fabrication Default) UART boot settings (Tx P0_4 and Rx P0_5) 20
Figure 14: DA14580/581Boot from external SPI memory 20
Figure 15: Data direction of UART within J4 20
Figure 16: DA14580/581 layout of headers J4 and J6 21
Figure 17: (Fabrication Default) DA14583 UART boot settings (Tx P0_4 and Rx P0_5) 22
Figure 18: Data direction of UART within J4 22
Figure 19: DA14583 Layout of headers J4 and J6 22
Figure 20: Block diagram with different memory locations 26
Figure 21: Setup of DA14580/581 DEVKT – BASIC during power measurement 34
Figure 22: Setup of DA14583 DEVKT – BASIC during power measurement 34
Figure 23: A general overview of the DA14581 development board 36
Figure 24: DA14580/581 Connection diagram of J10 and J11 36
Figure 25: A general overview of the DA14583 development board 37
Figure 26: DA14583 Connection diagram of J10 37
Figure 27: Open the CLI of SmartSnippets 38
Figure 28: SmartSnippets output 38
Figure 29: FTDI Latency Timer 42

Tables

Table 1: Content of the DEVKT – Basic Kit 5
Table 2: SDK Examples 7
Table 3: DA14580 9
Table 4: DA14581 9
Table 5: DA14583 10
Table 6: SPI connections 11
Table 7: Installation tools and drivers 23
Table 8: Way of Working (WoW) loading 'Blinky'-code 27
Table 9: Run an example on DA14580/581/583 29

 © 2015 Dialog Semiconductor

1 Terms and definitions

BLE	Bluetooth Low Energy
CS	Chip Select
DK	Development Kit
EEPROM	Electrically Erasable Programmable Memory
FTDI	Brand name of USB – UART interface
GPIO	General Purpose Input Output
OTP	One Time Programmable
PCB	printed circuit board
QFN	Quad-Flat No-leads
SDK	Software Development Kit
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SWD	Serial Wire Debug
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver/Transceiver
WLCSP	Wafer Level Chip Scale Packaging
WoW	Way of Working

2 References

1. DA14580, Datasheet, Dialog Semiconductor
2. DA14581, Datasheet, Dialog Semiconductor
3. UM-B-015, DA14580_581_583 Software Architecture, Dialog Semiconductor
4. DA14580 CB PXI QFN40 layout, Dialog Semiconductor
5. DA14580_CB_PXI_QFNP40, Dialog Semiconductor
6. DA14580_CB_PXI_WLCSP, Dialog Semiconductor
7. DA14580_CB_PXI_WLCSP_layout, Dialog Semiconductor
8. DA14580_MB_VB_layout, Dialog Semiconductor
9. DA14580_CB PXI_QFN48, Dialog Semiconductor
10. UM-B-005, DA14580_581_583 Peripheral Examples, Dialog Semiconductor
11. UM-B-010, DA14580_581_583 Proximity application, Dialog Semiconductor
12. AN-B-015 DA14580 Supply current measurement, Dialog Semiconductor

© 2015 Dialog Semiconductor

3 Introduction

DA14580/581/583 is a Bluetooth Smart chip, working with extremely low power while providing world-class RF performance, a small footprint and flexible peripheral configurations for a wide range of applications. The development kit includes a set of hardware (e.g. development board with on-board debugger), a Software Development Kit (SDK) (e.g. development tools, source code examples documents and so on) along with documentation. This document helps users to set up hardware/software development environment, install required software and quickly start product development with the help of example source code.

Web content can be downloaded at: www.dialog-semiconductor.com/support.

Product information about the DA14580, DA14581 and DA14583 can be found at: www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-da14580.

www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-development-tools/da14581-development-kit-basic.

3.1 Hardware content

In Figure 1 the kit components are shown and in Table 1 the parts are printed.

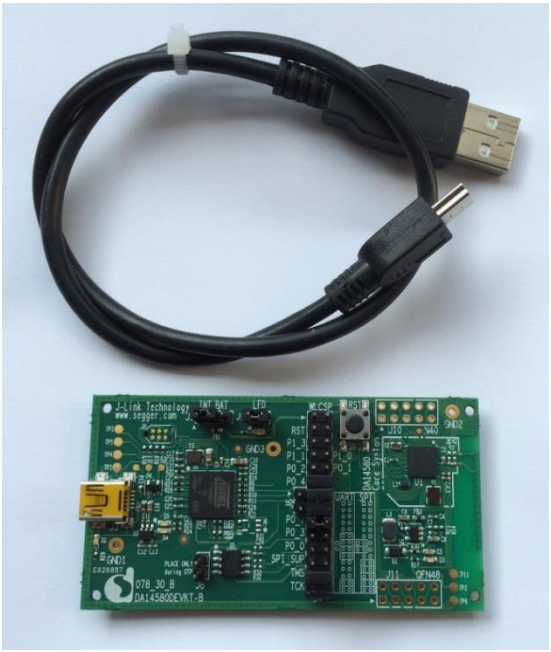


Figure 1: DEVKT – Basic Kit

Table 1: Content of the DEVKT – Basic Kit

DA14580/581/583 basic board
Battery: CR2032
USB → Mini USB Cable

Note 1 Kits are not pre-programmed so must be programmed before use!

© 2015 Dialog Semiconductor

3.2 Web content

3.2.1 Software Development Kit content

3.2.1.1 Tools

Web-link: www.dialog-semiconductor.com/support

SmartSnippets

SmartSnippets is a framework of PC based tools to control DA14580/581/583 development kit, consisting of

- OTP Programmer: Tool for OTP memory programming
- UART booter: Tool for downloading hex files to DA14580/581 SRAM over UART
- SPI & I2C memory programmer: Tool for SPI flash and I2C EEPROM programming

ConnectionManager

Connection Manager is a PC based software tool to control the link layer of the DA14580/581/583, with the following capabilities:

- Functional in Peripheral and Central role
- Set advertising parameters
- Set connection parameters
- Reading from Attribute database
- Perform production test commands

3.2.1.2 SDK documents

- UM-B-003, DA14580_581_583 Software development guide
- UM-B-004, DA14580_581_583 Peripheral drivers
- UM-B-005, DA14580_581_583 Peripheral examples
- UM-B-006, DA14580 Sleep mode configuration
- UM-B-007, DA14580 Software Patching over the Air (SPOTA)
- UM-B-008, DA14580_581_583 Production test tool
- UM-B-010, DA14580_581_583 Proximity application
- UM-B-011, DA14580 Memory map – scatter file
- UM-B-012, DA14580 Secondary boot loader
- UM-B-013, DA14580 External Processor Interface over SPI
- UM-B-014, DA14580_581 Development Kit
- UM-B-015, DA14580_581_583 Software architecture

© 2015 Dialog Semiconductor

3.2.1.3 SDK source code examples (created with Keil IDE)

Web-link:

- **dk_apps.** This folder holds all the necessary folders needed for DA14580/581 application development.
  - **dk_apps\keil_projects\proximity**

    The folder contains the following subfolders and in each one of them resides the respective project file:

Table 2: SDK Examples

Folder	Project File	Description
prox_monitor_ext\Keil_4 **prox_monitor_ext\Keil_5**	prox_monitor_ext.uvproj prox_monitor_ext_581.uvproj prox_monitor_ext_583.uvproj prox_monitor_ext.uvprojx prox_monitor_ext_581.uvprojx prox_monitor_ext_583.uvprojx	Proximity Monitor (External processor configuration)
prox_reporter_ext\Keil_4 **prox_reporter_ext\Keil_5**	prox_reporter_ext.uvproj prox_reporter_ext_581.uvproj prox_reporter_ext_583.uvproj prox_reporter_ext.uvprojx prox_reporter_ext_581.uvprojx prox_reporter_ext_583.uvprojx	Proximity Reporter (External processor configuration)
prox_reporter\Keil_4 **prox_reporter\Keil_5**	prox_reporter.uvproj prox_reporter_581.uvproj prox_reporter_583.uvproj prox_reporter.uvprojx prox_reporter_581.uvprojx prox_reporter_583.uvprojx	Proximity Reporter (Integrated processor configuration)
prox_monitor_ext_usb\Keil_4 **prox_monitor_ext_usb\Keil_5**	prox_monitor_ext_usb.uvproj prox_monitor_ext_usb_581.uvproj prox_monitor_ext_usb_583.uvproj prox_monitor_ext_usb.uvprojx prox_monitor_ext_usb_581.uvprojx prox_monitor_ext_usb_583.uvprojx	Proximity Monitor (External processor configuration) Version for USB dongle
prox_reporter_ext_usb\Keil_4 **prox_reporter_ext_usb\Keil_5**	prox_reporter_ext_usb.uvproj prox_reporter_ext_usb _581.uvproj prox_reporter_ext_usb _583.uvproj prox_reporter_ext_usb.uvprojx prox_reporter_ext_usb _581.uvprojx prox_reporter_ext_usb _583.uvprojx	Proximity Reporter (External processor configuration) Version for USB dongle
prox_reporter_ext_spi\Keil_4 **prox_reporter_ext_spi\Keil_5**	prox_reporter_ext_spi.uvproj prox_reporter_ext_spi_581.uvproj prox_reporter_ext_spi_583.uvproj prox_reporter_ext_spi.uvprojx prox_reporter_ext_spi_581.uvprojx prox_reporter_ext_spi_583.uvprojx	Proximity Reporter (External processor) SPI version

© 2015 Dialog Semiconductor

- **dk_apps\keil_projects\prod_test:** This folder includes the source code of the production test firmware. Refer to UM-B-008_DA14580_581_583_Production_test_tool.pdf for more information how to build and use it.

- **host_apps:** This folder holds the DA14580/581/583 PC applications:
  - ***host_apps\windows\proximity:*** This folder includes two Windows C applications, with each one acting as part of a proximity monitor and a proximity reporter application. They are placed in subfolders *monitor* and *reporter* respectively. For details, please read the *DA14580 Proximity Application* Guide.
  - **host_binaries\windows\proximity:** This folder includes two pre-compiled Windows executables which correspond to the C applications described right above and are included for user convenience.
  - **peripheral_examples:** This folder includes sample code of how to use the peripheral blocks of the DA14580 (e.g. UART, SPI, I2C etc.) bundled to a demo-kit. For details, please refer to [9].

- **tools:**
  - **tools\prod_test\prod_test_cmds:** This folder includes the source code of the production test tool. Refer to UM-B-008_DA14580_581_583_Production_test_tool.pdf for more information how to build and use it.

© 2015 Dialog Semiconductor

3.3 DA14580 – DA14581 basic kit

3.3.1 Differences between the DA14580/580/583

The motherboard of the DEVKT – Pro will not change. Only the daughters of the DA14580/581/583 are different.

Table 3: DA14580

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
WL-CSP34	ROM 84kBytes OTP 32kBytes RAM 50kBytes	12	2.5x2.5x0.5mm, pitch 0.4mm	• Bluetooth 4.0 + 4.1 • Cortex M0 application processor • Power supply 0.9 - 3.3V • Single pin RF I/O • Rich set of analog and digital peripherals	Beacon & Proximity Health & Fitness HID Smart Home
QFN40		24	5x5x0.9mm, pitch 0.4mm		
QFN48		32	6x6x0.9mm, pitch 0.4mm		

Table 4: DA14581

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
WL-CSP34	ROM 84kBytes OTP 32kBytes RAM 50kBytes	12	2.5x2.5x0.5mm, pitch 0.4mm	• Bluetooth 4.0 + 4.1 • Cortex M0 application processor • Power supply 0.9 - 3.3V • Single pin RF I/O • Rich set of analog and digital peripherals • 8 connections Optimized boot time	Wireless charging (A4WP) HCI
QFN40		24	5x5x0.9mm, pitch 0.4mm		

© 2015 Dialog Semiconductor

Table 5: DA14583

Product	Memory size	General Purpose I\O's	Package	Key Features	Applications
QFN40	**Flash** 1 Mbit (128kBytes) ROM 84kBytes OTP 32kBytes RAM 50kBytes	24	5x5x0.9mm, pitch 0.4mm	• Bluetooth 4.0 + 4.1 • Cortex M0 application processor • Power supply 2.35 - 3.3V • Single pin RF I/O • Rich set of analog and digital peripherals	Beacon & Proximity Health & Fitness HID Smart Home

Note 1 The only hardware difference between the DEVKT -Basic 14580 and the DEFKT -Basic 14581, is the design in of QFN48 (580) and the QFN40 (580/581). The silkscreen may have small textual differences.

Remark: DA14583 will run in BUCK mode only!

© 2015 Dialog Semiconductor

3.4 Pinning

In Figure 2 the pinout of the DA14583 is shown. New, compared to the DA14580/581, are the connections to the internal SPI flash memory.

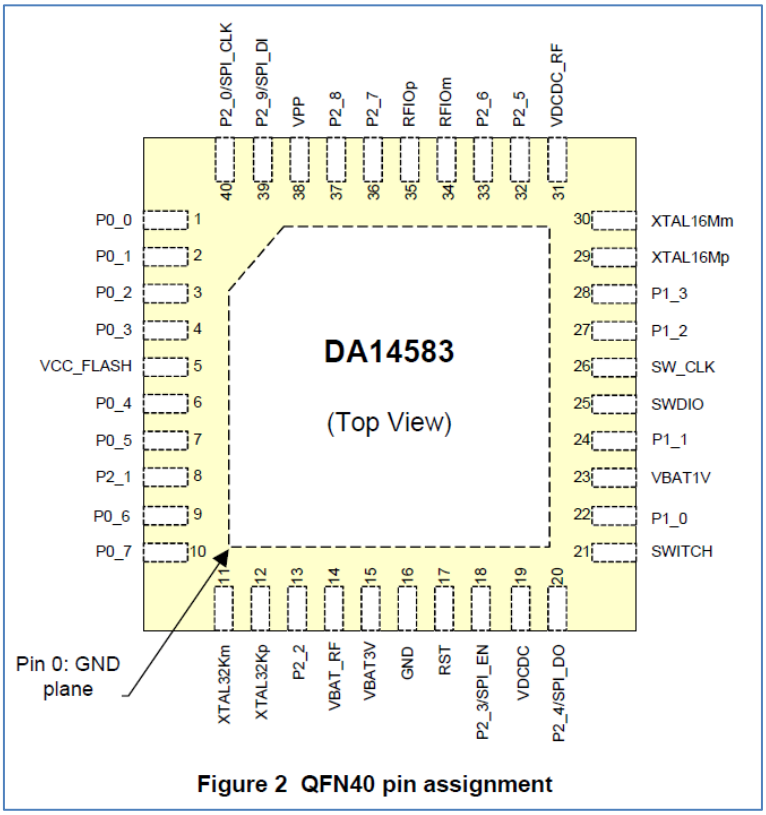


Figure 2 QFN40 pin assignment

Figure 2: QFN40 pin assignment from datasheet

Table 6: SPI connections

port DA14583	function	remark
P2_0	SPI_CLK	SCLK (Note 2)
P2_9	SPI_DI	MOSI (Note 2)
P2_4	SPI_DO	MISO (Note 2)
P2_3	SPI_EN	not to be used for external SPI (!)
	VCC_FLASH	power for internal Flash Memory
	GND	

Note 2 shared with internal flash memory

When external SPI components are used, SPI_EN is occupied for internal use. Another pin should be chosen for SPI_EN of the external component.

By using a Secondary Bootloader the proper pins are programmed to load the booting software from the SPI-memory at startup.

© 2015 Dialog Semiconductor

3.4.1 PCB design and functionalities

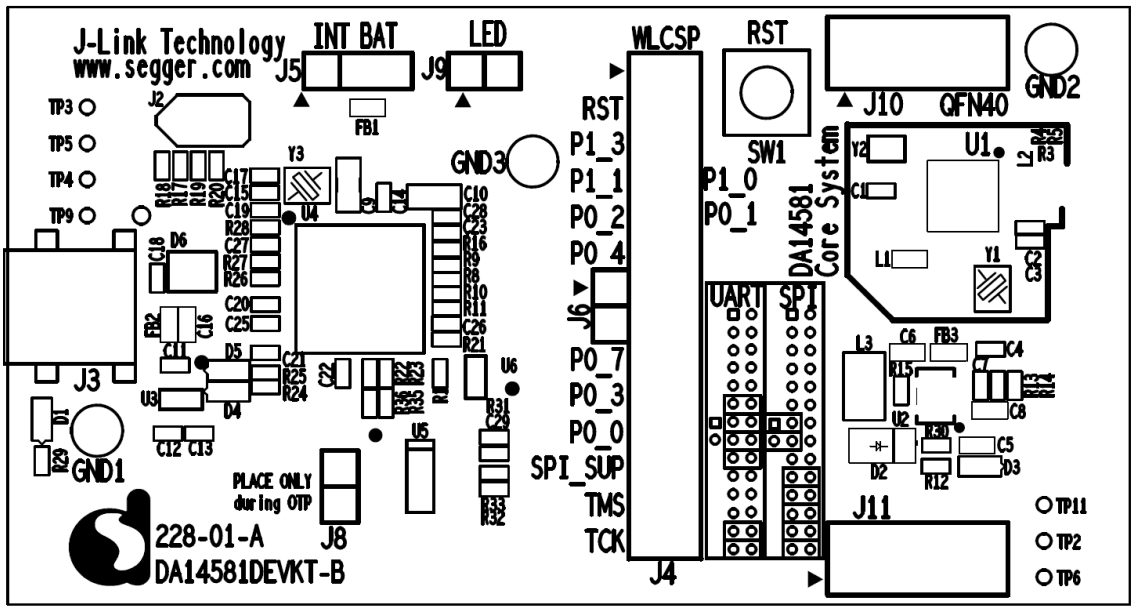


Figure 3: Components on PCB top layer of the DA14580/581Basic kit

The different components and functionalities are shown in Figure 4.
A larger picture is shown in Appendix A

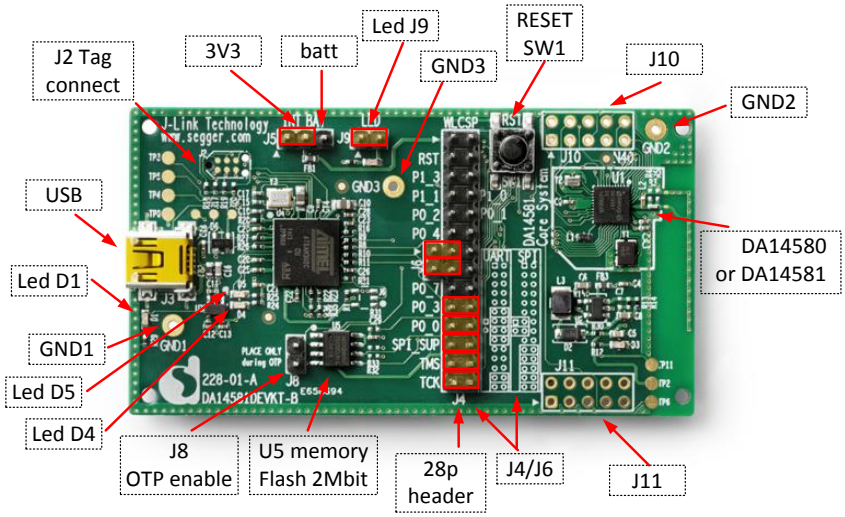


Figure 4: Topview of PCB DA14580/581 with components and functionalities

© 2015 Dialog Semiconductor

The top-screen layer of the DA14583 Basic Kit PCB is shown in Figure 5.

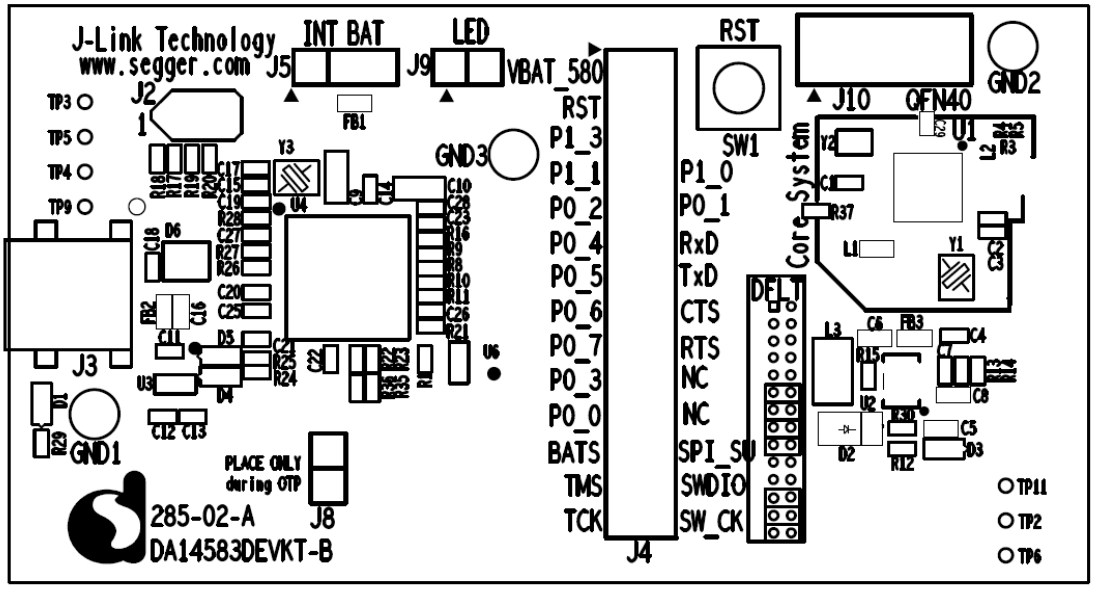


Figure 5: Components on PCB top layer of the DA14583 Basic kit

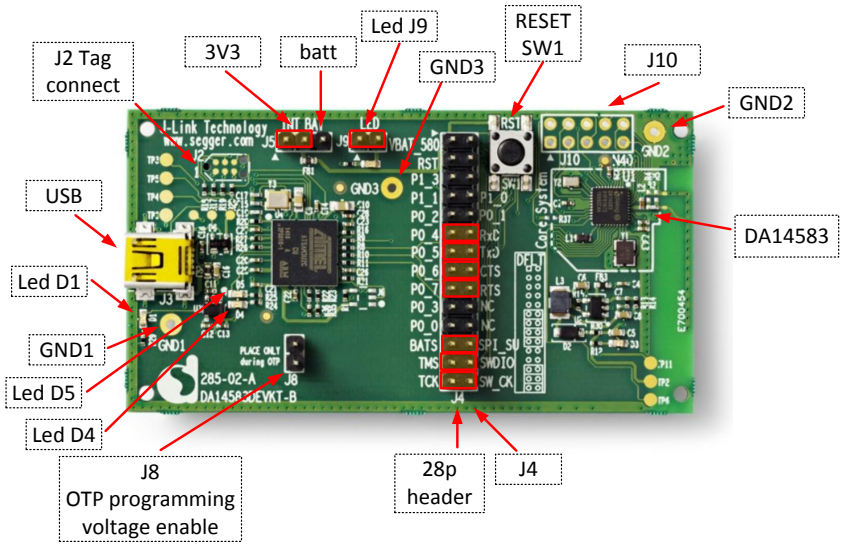


Figure 6: Topview of PCB DA14583 with components and functionalities

© 2015 Dialog Semiconductor

3.4.2 Schematics and layouts

3.4.2.1 DA1580DEVKT-B_va layout

For the schematics and layout of the board in full detail, please refer to the respective documents on the portal. See link: www.dialog-semiconductor.com/support.

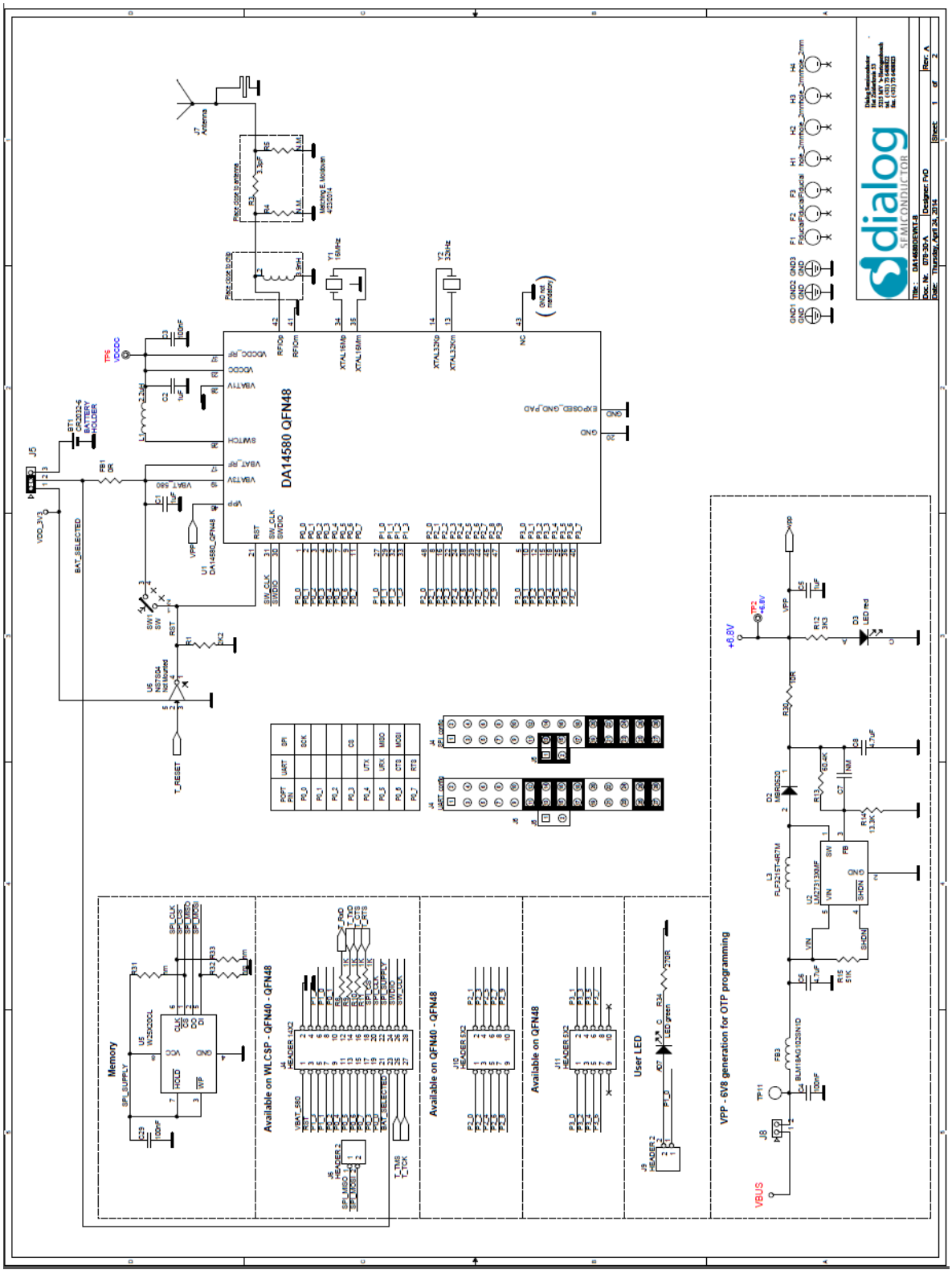


Figure 7: DA14580DEVKT-B_vb part 1

© 2015 Dialog Semiconductor

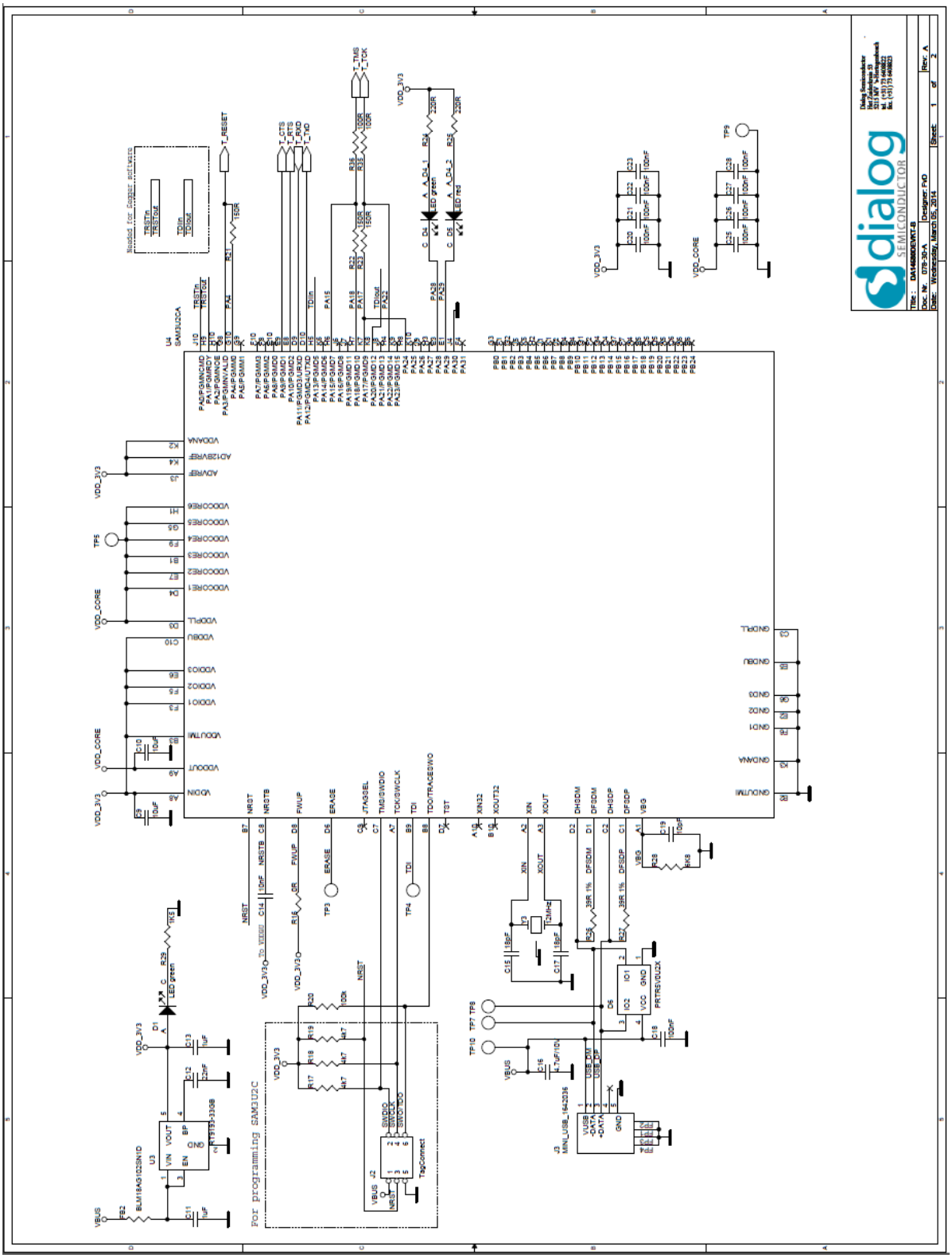


Figure 8: DA14580DEVKT-B_vb part 2

© 2015 Dialog Semiconductor

3.4.2.2 DA1581DEVKT-B_va layout

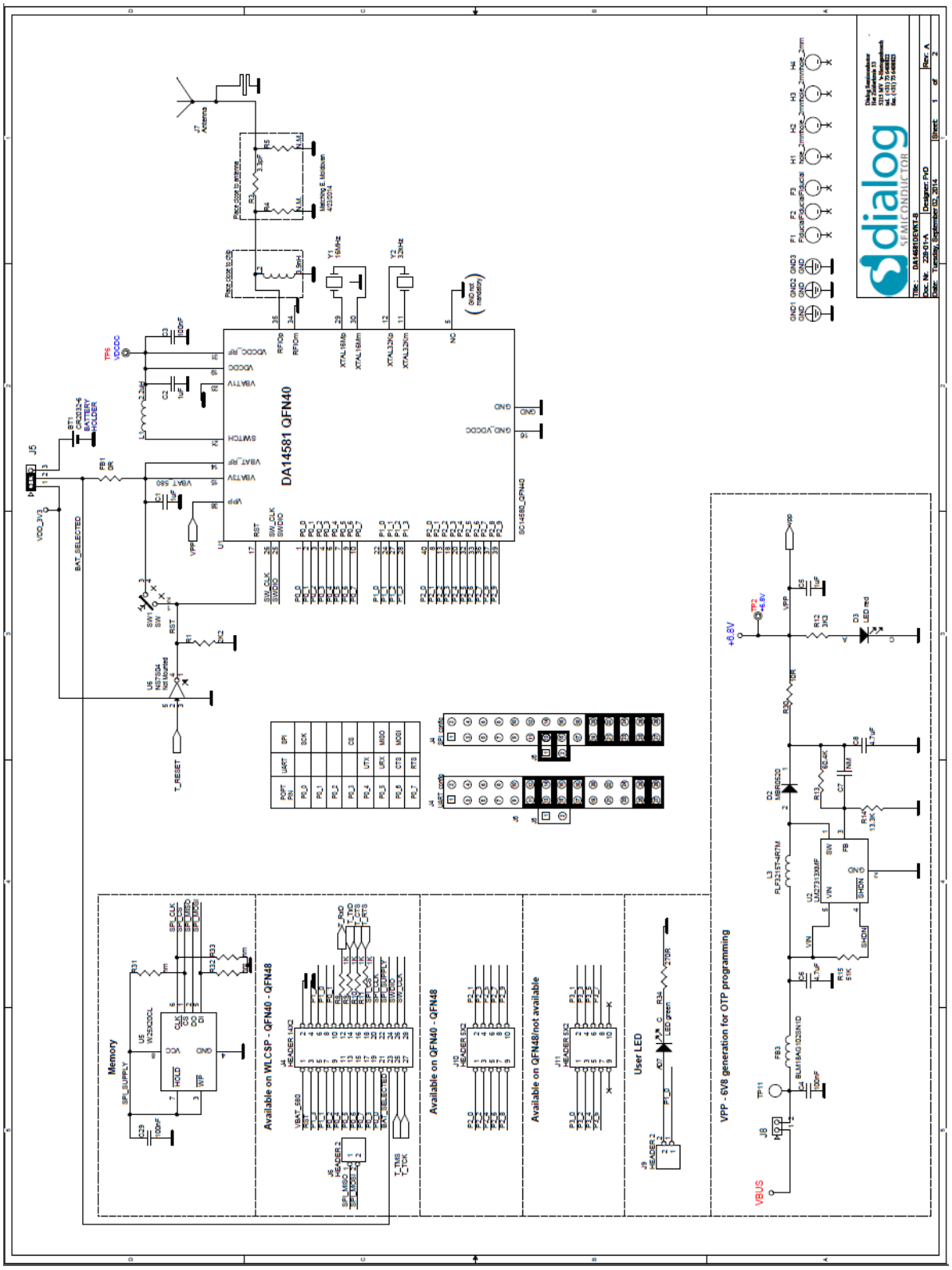

DA14581 QFN40
Memory
Available on WLCSP - QFN40 - QFN48
Available on QFN40 - QFN48
Available on QFN48/not available
User LED
VPP - 6V8 generation for OTP programming

Figure 9: DA1581DEVKT-B_va part 1

© 2015 Dialog Semiconductor

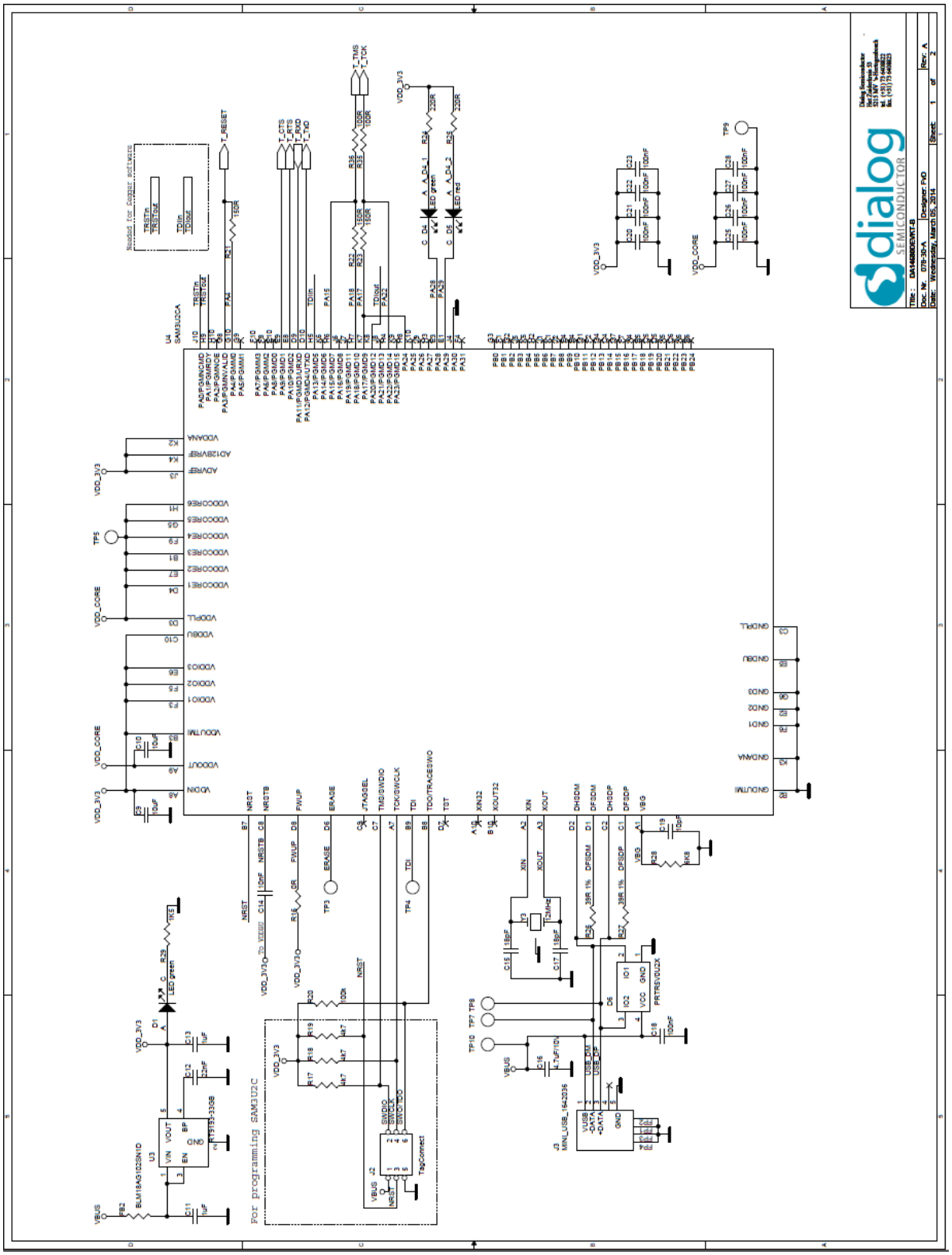


Figure 10: DA1581DEVKT-B_va part 2

© 2015 Dialog Semiconductor

3.4.2.3 DA1583DEVKT-B_va layout

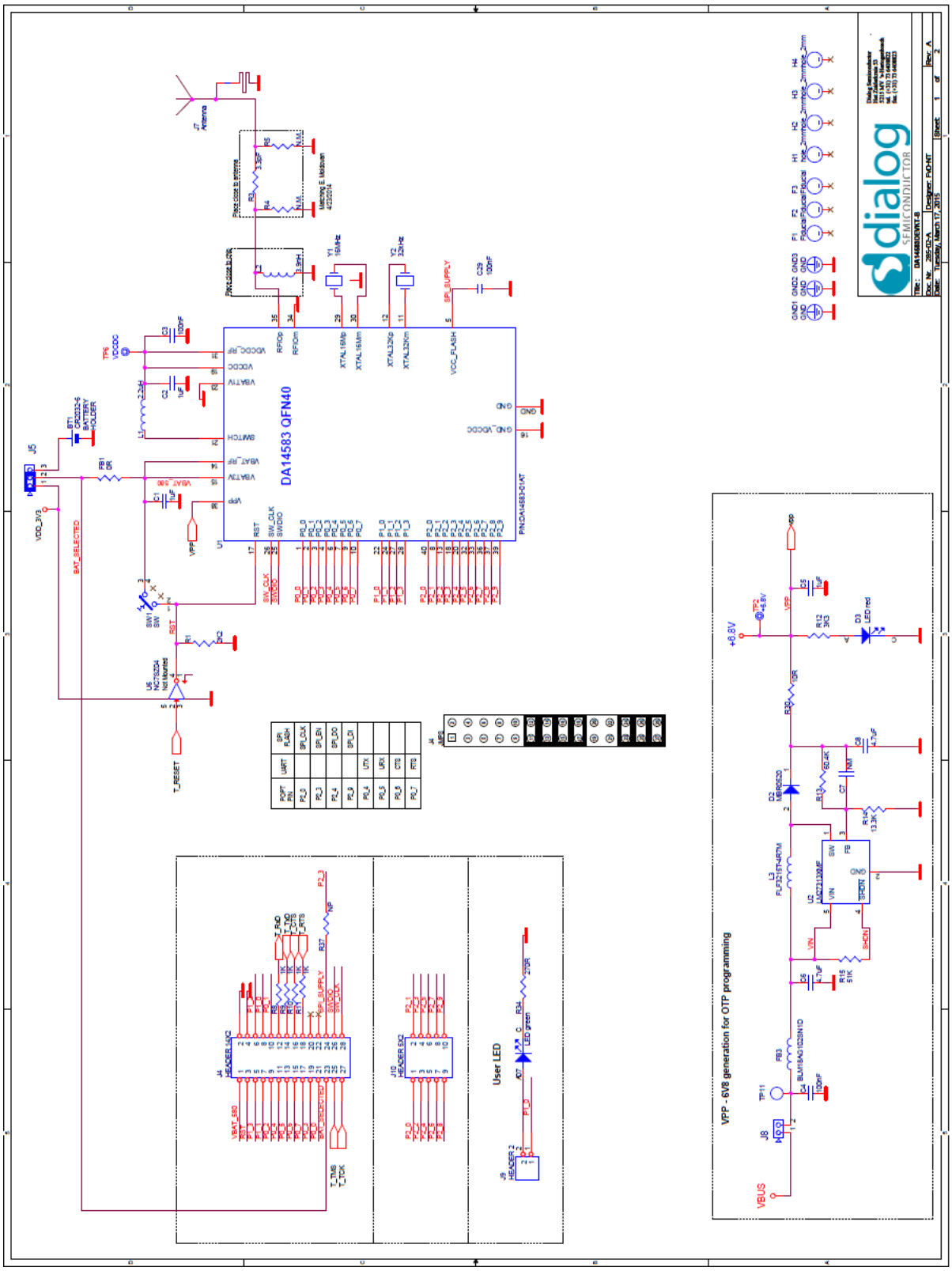


Figure 11: DA1583DEVKT-B_va part 1

© 2015 Dialog Semiconductor

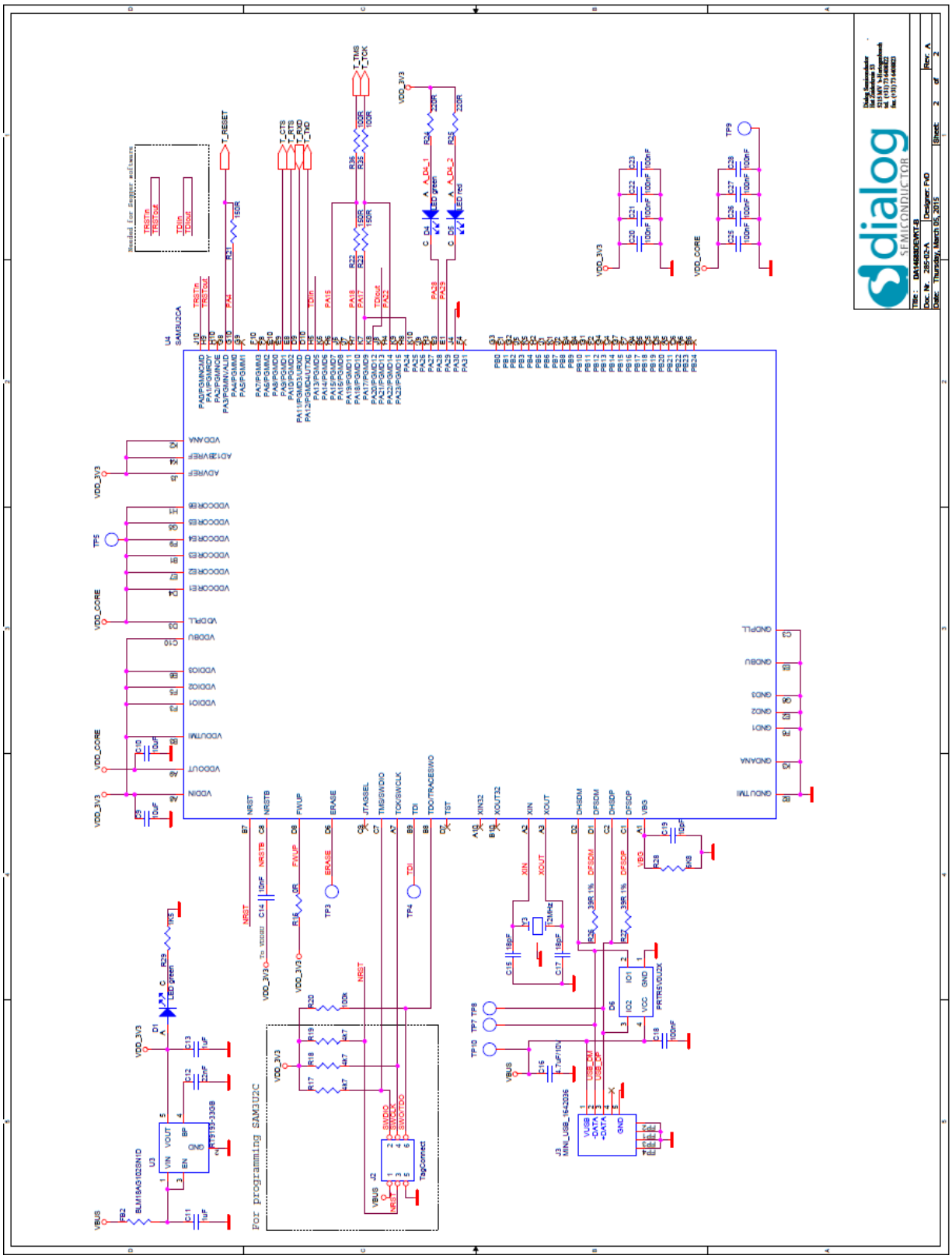


Figure 12: DA1583DEVKT-B_va part 2

© 2015 Dialog Semiconductor

3.4.3 Configuring the DA14580/581 basic kit-board by jumper settings

Different functionalities are shown in Appendix A, and the jumper settings are displayed below.

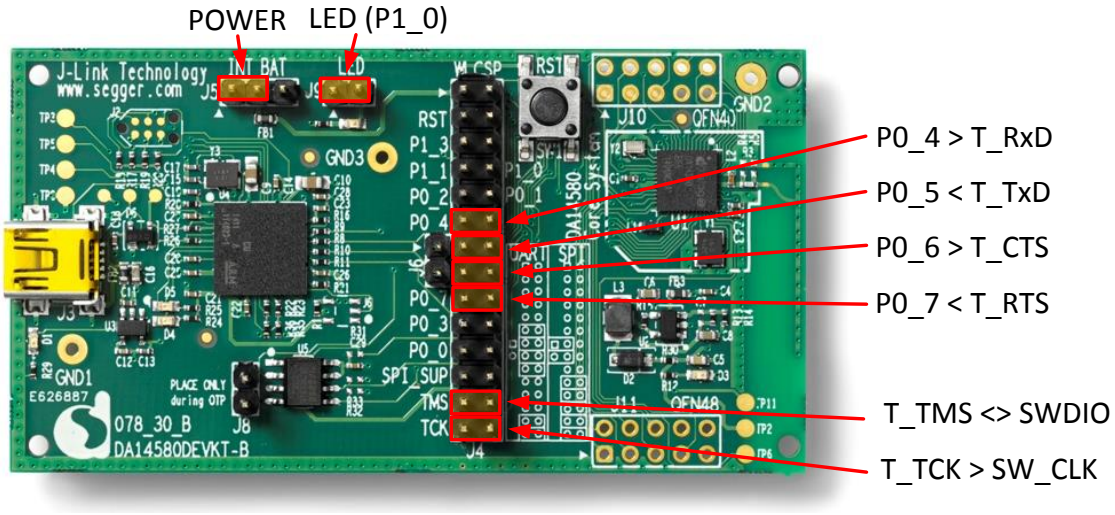


Figure 13: DA14580/581 (Fabrication Default) UART boot settings (Tx P0_4 and Rx P0_5)

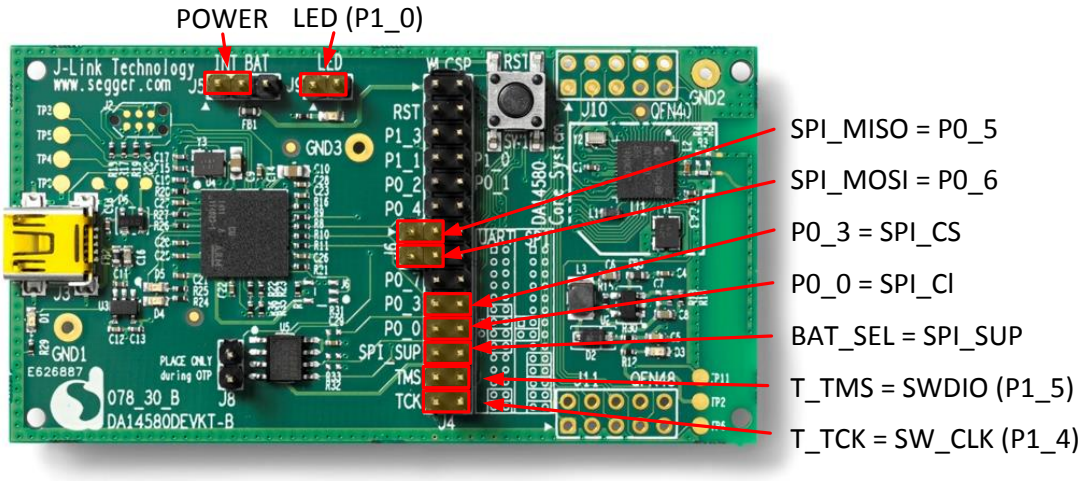


Figure 14: DA14580/581Boot from external SPI memory

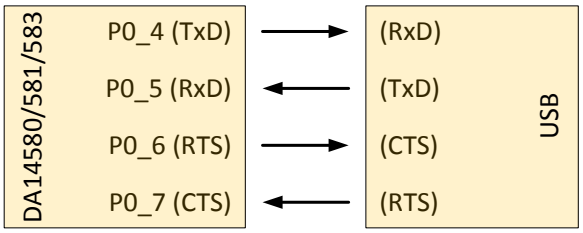


Figure 15: Data direction of UART within J4

© 2015 Dialog Semiconductor

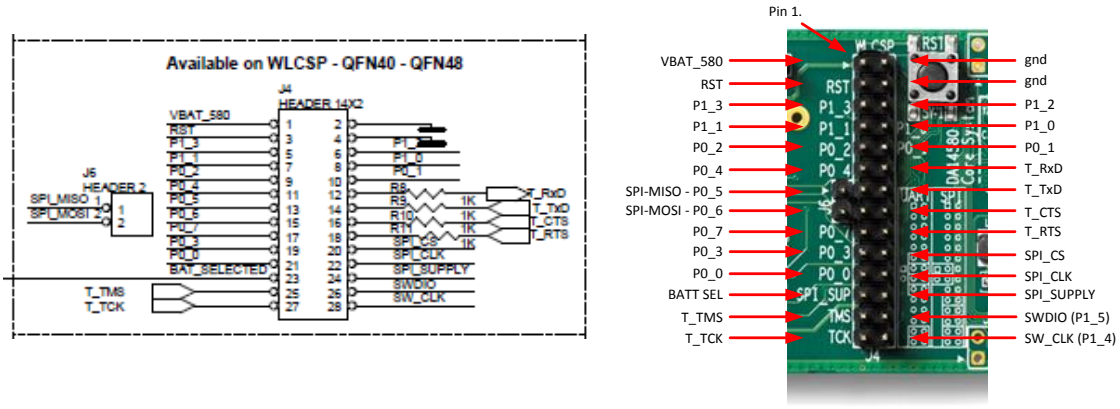


Figure 16: DA14580/581 layout of headers J4 and J6

Example: when jumper J4 (27-28) is placed, connection T_CK = SW_CLK is made.

On this board only the Buck mode is used. A choice can be made between 3V3 (via USB) (J5 1-2) or Vdd (via a coin cell) (J5 2-3). The battery (coin cell) is placed in the battery-socket on the back of the PCB.
No battery is needed when running via the USB-mini-cable.

Note 1 The DA14580 is equipped with a QFN48 device and the DA14581 has a QFN40 die mounted.

© 2015 Dialog Semiconductor

3.4.4 Configuring the DA14583 basic kit-board by jumper settings

Different functionalities are shown in Appendix A, and the jumper settings are displayed below.

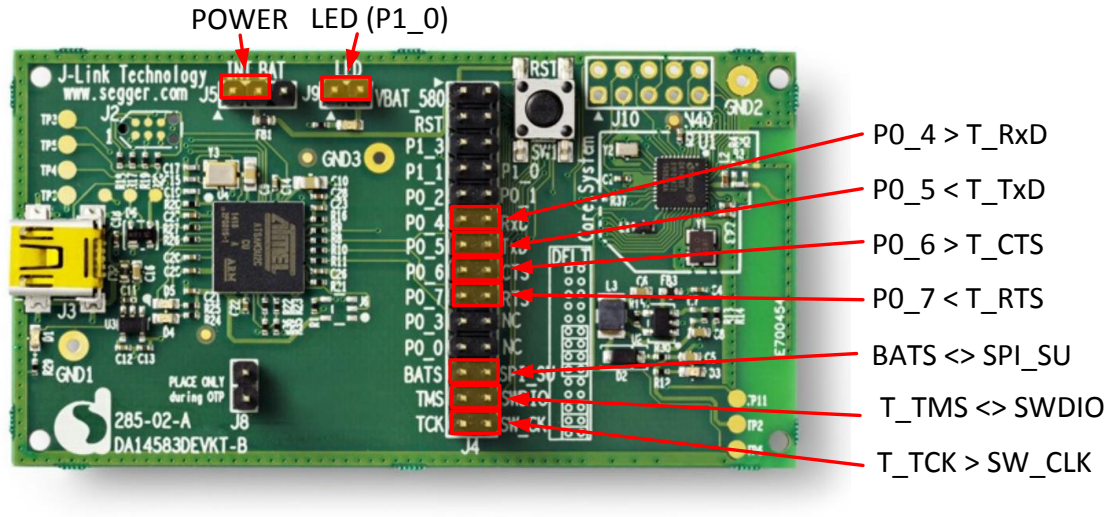


Figure 17: (Fabrication Default) DA14583 UART boot settings (Tx P0_4 and Rx P0_5)

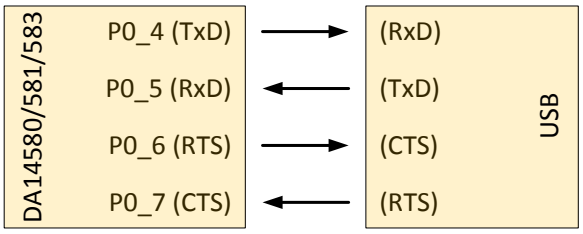


Figure 18: Data direction of UART within J4

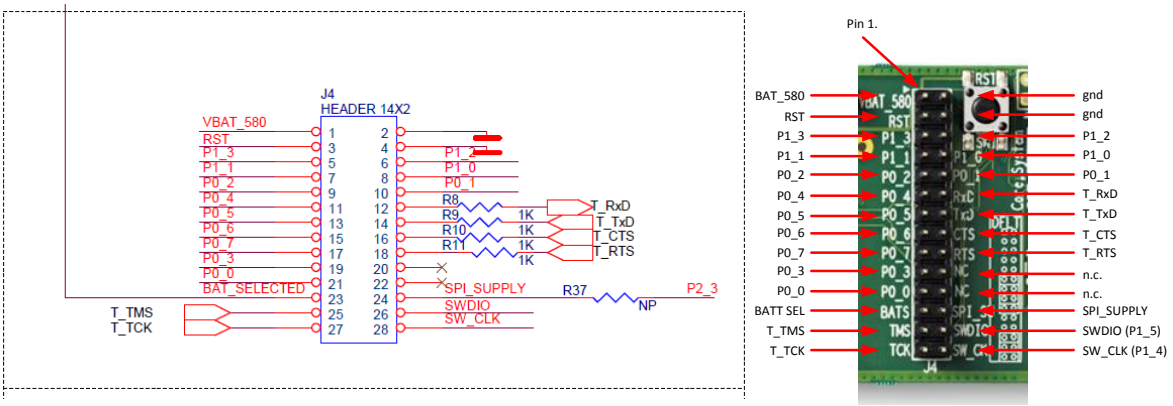


Figure 19: DA14583 Layout of headers J4 and J6

Example: when jumper J4 (27-28) is placed, connection T_CK = SW_CLK is made.

On this board only the Buck mode is used. A choice can be made between 3V3 (via USB) (J5 1-2) or Vdd (via a coin cell) (J5 2-3). The battery (coin cell) is placed in the battery-socket on the back of the PCB.
No battery is needed when running via the USB-mini-cable.

Note 1 The DA14583 is equipped with a QFN40 device.

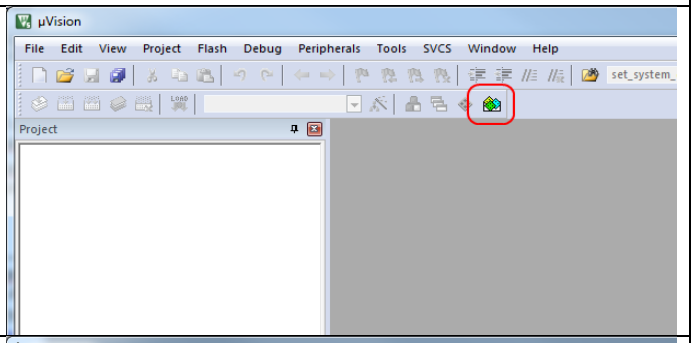
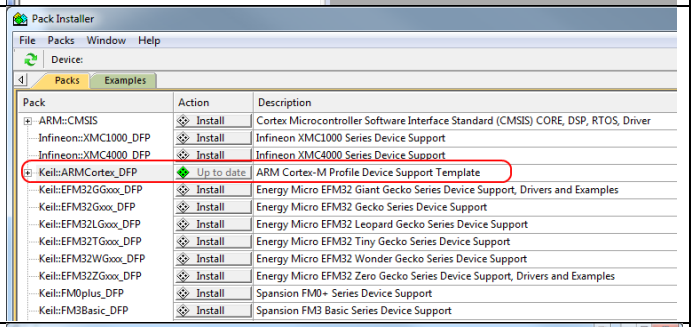
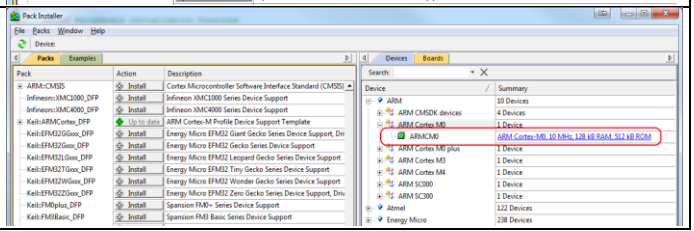
I

© 2015 Dialog Semiconductor

4 Installation of tools and drivers

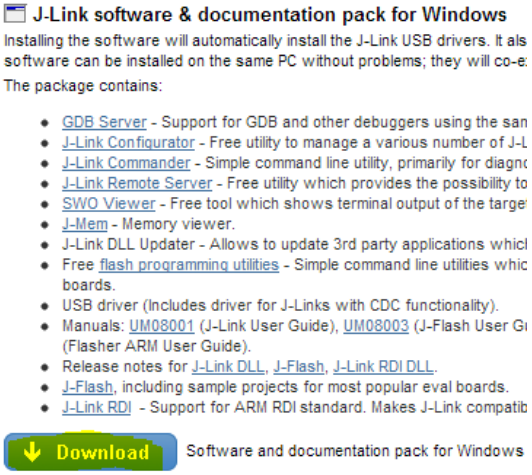
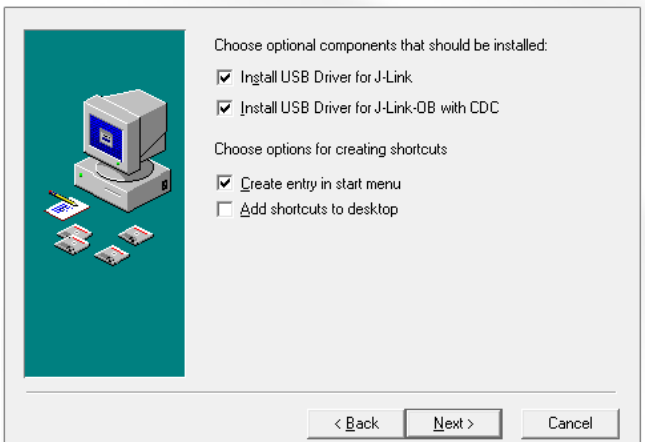
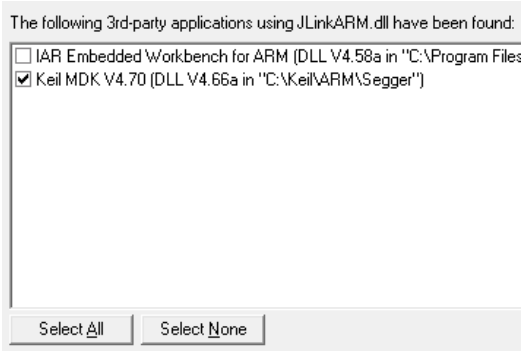
To install the Software development environment, please follow the steps as shown in Table 7.

Table 7: Installation tools and drivers

4.1 Keil		
4.1.1	Download and install Keil MDK-ARM uVision IDE.	Keil: https://www.keil.com/download/product/ Keil MDK-ARM Version 5 – Installation: http://www2.keil.com/mdk5/install Starters Guide: http://www.keil.com/uvision/ide_ov_starting.asp
4.1.2	For uVision Version you have to install the ARM Cortex M profile package (see also http://www.keil.com/dd2/Pack/)	
4.1.3	You should see a list of packs as shown on the right. If you do not see this list, please click the “Packs” menu item and select the “Check for Updates” option to download an updated list. Click on the “Install” button to the right of “Keil::ARMCortex_DFP” package.	
4.1.4	If the installation is successful, the pack installer window should look like this.	

 © 2015 Dialog Semiconductor

4.2 SEGGER Jlink driver

4.2.1	Download and install the Jlink software & documentation pack for Windows. *Please note that your SEGGER Jlink serial number is required for downloading.* *Use sticker-number on backside of board.* .	 http://www.segger.com/jlink-software.html
4.2.2	In order to have the USB controller properly recognized by Windows as a J-Link device, you have to install the driver with the settings shown in the side figure.	
4.2.3	At the end of the installation, please tick the IDE (Keil MDK Vxx) that you are using.	

4.3 FTDI driver

4.3.1	The Development Kit uses the D2xx driver. For Windows, this driver is part of the Combined Driver Model (CDM)	USB Drivers:

© 2015 Dialog Semiconductor

	driver. (It is recommended that the latest driver available from the FTDI page is used.) **Latency Timer:** see Appendix H	http://www.ftdichip.com/Drivers/D2XX.htm Installation Guide: http://www.ftdichip.com/Support/Documents/InstallGuides.htm
4.4 Tera Term		
4.4.1	Download and install Tera Term on your PC.	Tera Term: http://en.sourceforge.jp/projects/ttssh2/releases/

© 2015 Dialog Semiconductor

5 Quick start: Downloading software

The DEVKT –Basic is equipped with, on the chip, SRAM (42k) and OTP (32k). Mounted on the board is external SPI flash memory (2Mbit).

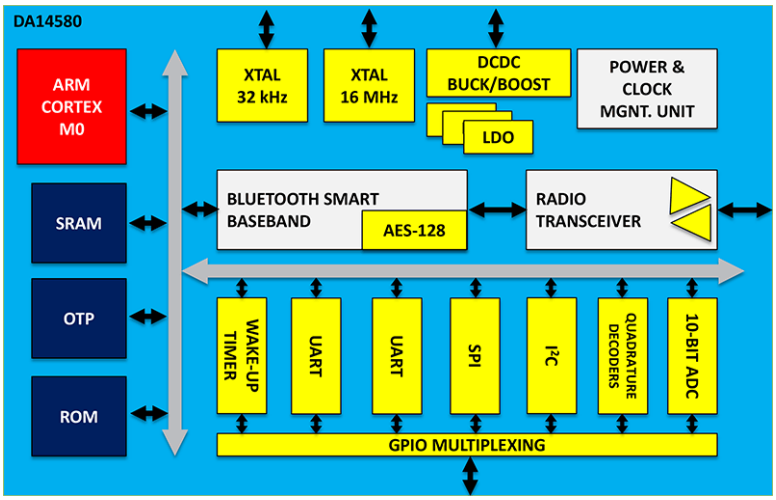


Figure 20: Block diagram with different memory locations

Software can be downloaded to:

- SRAM
  - Keil IDE
  - SmartSnippets
  - Command Line Interface (CLI)
  - Connection Manager
- OTP
  - SmartSnippets
  - CLI
- SPI (flash)
  - SmartSnippets
  - CLI

Example: loading software (hex-file) by using SmartSnippets

- PC -> UART -> DA14580/581/583
- PC -> UART -> DA14580/581/583 -> SPI (flash)
- PC -> UART -> DA14580/581/583 -> OTP

For the settings of the jumpers see Figure 13.

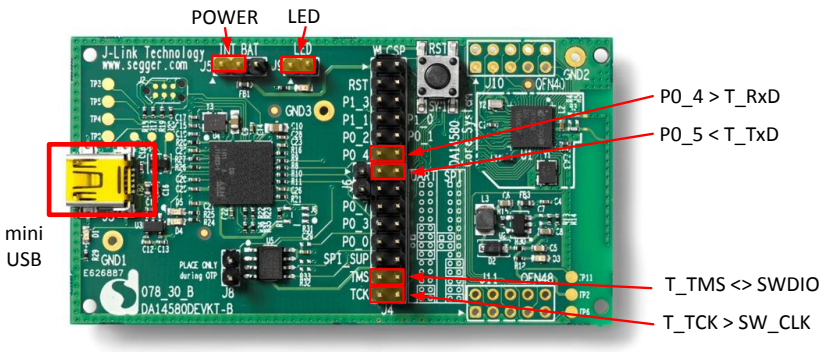
© 2015 Dialog Semiconductor

5.1 Software downloaded via Keil IDE: USB (via SWD) to SRAM

Way of working: the Keil IDE (uVision 5.12) is used to load software into the board. Software is downloaded to SRAM through SWD (Serial Wire Debug). After downloading the software (see 4.1), the software can be debugged via the same IDE.

Table 8: Way of Working (WoW) loading 'Blinky'-code

<table>
<tr><th colspan="3">Preparations</th></tr>
<tr><td colspan="3">This chapter shows the user how to quickly set up the software development environment of the BLE (Bluetooth Low Energy) demo</td></tr>
<tr><td>5.1.1</td><td>Attach mini USB cable to PC and demo board.
Functionality of this cable:
● power
● programming
● debugging
● UART

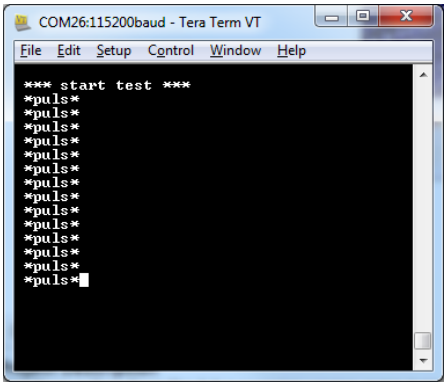
The DEVKT –B 581 and DEVKT –B 583 can be programmed in the same way.</td><td> 
</td></tr>
</table>

© 2015 Dialog Semiconductor

5.1.2	Add 4 jumpers, as shown in the photo of 5.1.1.	Jumpers: J5 (1-2) 'POWER' = left-side, J9 'LED', J4 (11-12) 'P0_4 > T_RxD', J4 (13-14) 'P0_5 < T_TxD', 'TMS' - J4 (25-26) 'T_TMS <> SWDIO' and 'TCK' - J4 (27-28) 'T_TCK > SW_CLK'.

5.1.3	Start Keil IDE software and load 'Blinky' example code. Via this Blinky-code the LED on the board starts blinking. Also added in this code, is the UART sending a text string.
	See 0
		/**
		**
		*
		* @file main.c
		*
		* @brief Blinky example for DA14580/581 SDK
		*
		* Copyright (C) 2012. Dialog Semiconductor Ltd, unpublished work. This computer
		* program includes Confidential, Proprietary Information and is a Trade Secret of
		* Dialog Semiconductor Ltd. All use, disclosure, and/or reproduction is prohibited
		* unless authorized in writing. All Rights Reserved.
		*
		* <bluetooth.support@diasemi.com> and contributors.
		*
		**
		*/
		#include <stdio.h>
		
		#include "global_io.h"
		#include "common_uart.h"
		#include "periph_setup.h"
		#include "gpio.h"
		
		#define LED_OFF_THRESHOLD 10000
		#define LED_ON_THRESHOLD 400000
		
		void system_init(void);
		void blinky_test(void);
		
		/**
		**
		* @brief Main routine of the UART example
		*
		**
		*/
		int main (void)
		{
		system_init();
		periph_init();
		blinky_test();
		while(1);
		}

5.1.4	In file 'periph_setup.h' default **P0_4** Tx and **P0.5** Rx are used as UART IO-ports.	// Select UART settings
		#define UART2_BAUDRATE UART_BAUDRATE_115K2
		#define UART2_DATALENGTH UART_CHARFORMAT_8
		#define UART2_PARITY UART_PARITY_NONE
		#define UART2_STOPBITS UART_STOPBITS_1
		#define UART2_FLOWCONTROL UART_FLOWCONTROL_DISABLED
		
		#define UART2_GPIO_PORT GPIO_PORT_0
		#define UART2_TX_PIN GPIO_PIN_4
		#define UART2_RX_PIN GPIO_PIN_5
		#define UART2_ENABLED
		// LED
		#define LED_PORT GPIO_PORT_1
		#define LED_PIN GPIO_PIN_0

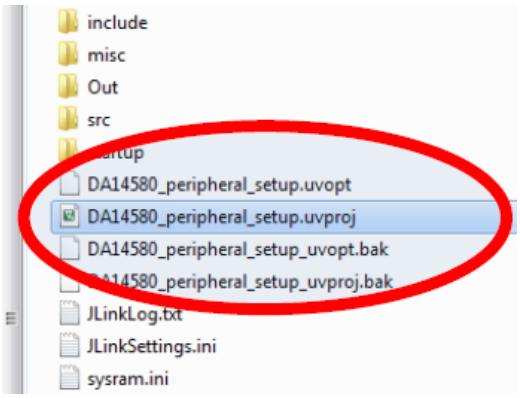
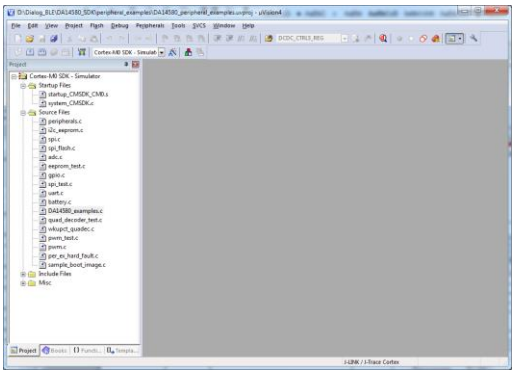
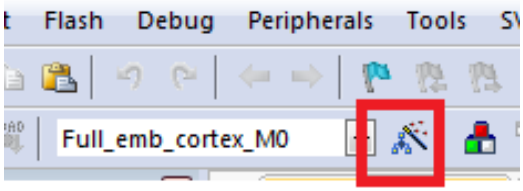
5.1.5	Compile and Debug the software. The blinking led D7 is visible and the TeraTerm screen is showing 'popping up' lines. Choose the right com-port via the Device Manager of your PC.	

© 2015 Dialog Semiconductor

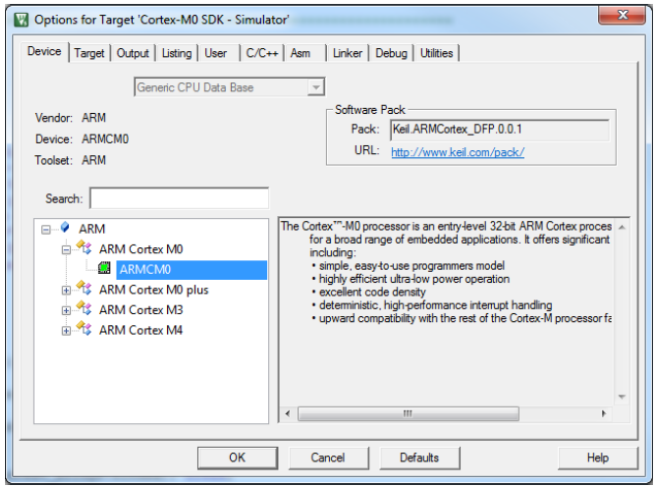
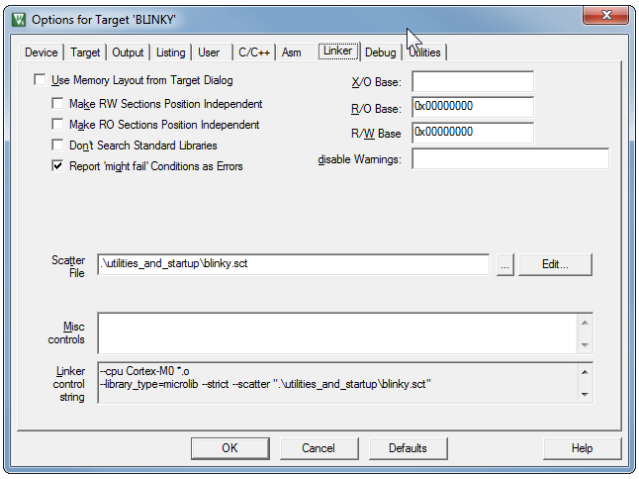
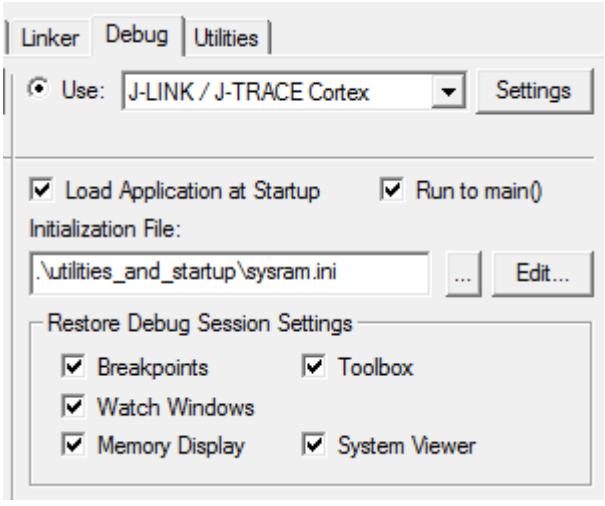
6 Using the demo kit

Follow these steps shown in Table 9 to easily create a working demo kit.

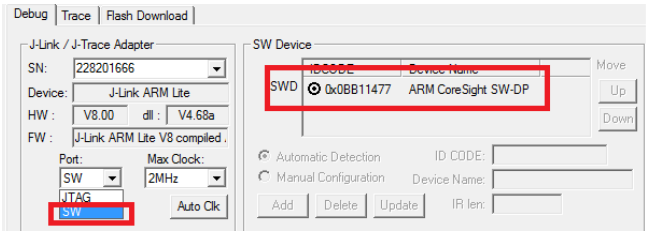
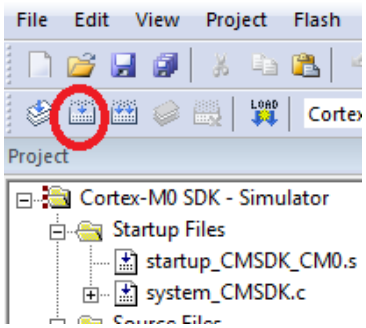
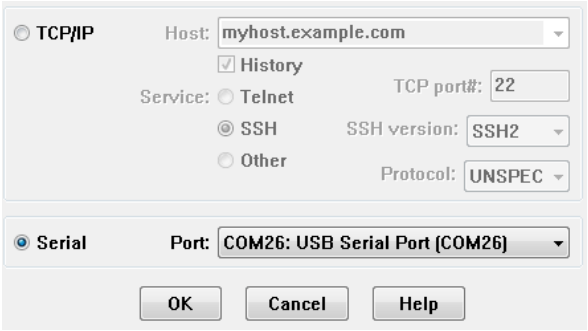
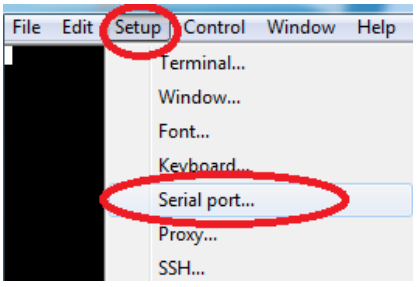
Table 9: Run an example on DA14580/581/583

6.1 Run an example on the DA14580/581/583		
6.1.1	After you download the SDK at www.dialog-semiconductor.com/support The source code example can be found in the example directory called "*peripheral_examples*\". Double click "580_peripheral_setup.**uvproj**	
6.1.2	The development environment should look like this when the project is opened with Keil.	
6.1.3	Click on the "Target Options" button	

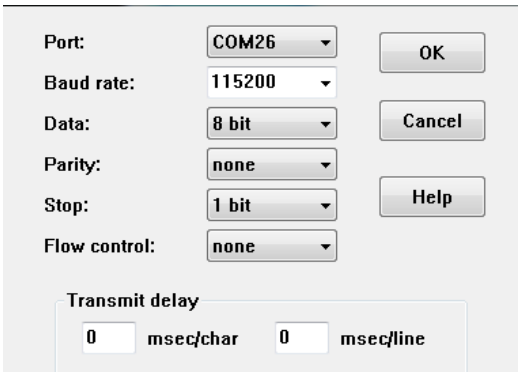
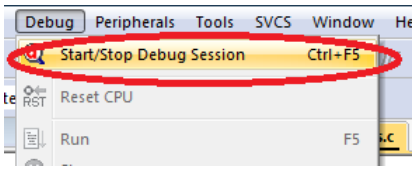
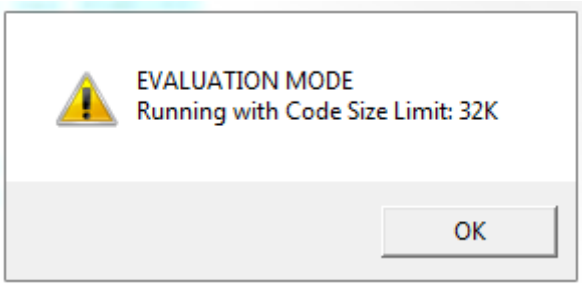
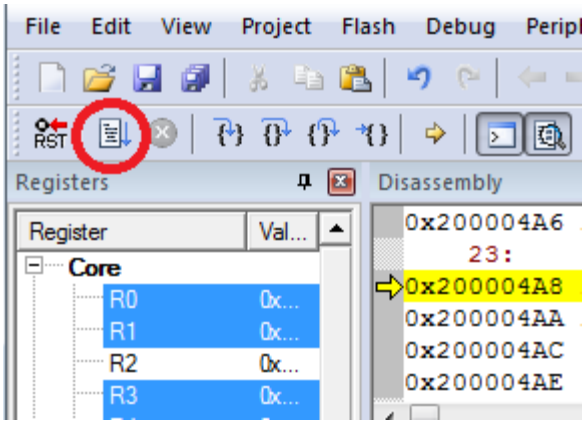
© 2015 Dialog Semiconductor

6.1.4	'Options for Target' → 'Device' -screen should look like this.	
6.1.5	Scatterfiles (.sct) are used for selecting memory areas.	 scatter file selection in 'Options for Target'
6.1.6	Make sure "J-LINK/J-Trace Cortex" is selected as shown and the initialization file field is set correctly to ".\sysram.ini". Click on "**Settings**" for next screen.	

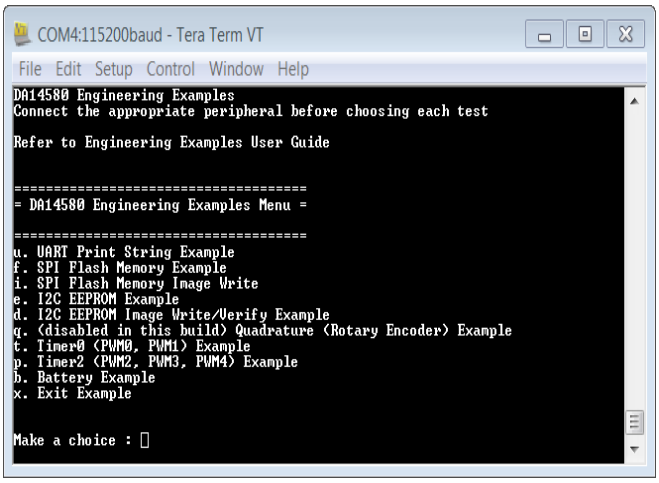
 © 2015 Dialog Semiconductor

6.1.7	After clicking the “**Settings**”-button above, make sure the SW Device has been detected correctly.	
6.1.8	Click “**OK**” to save the settings.	All settings have been saved properly now, and you can continue to build the example.
6.1.9	Build the project by pressing “F7” key, or click the build button as shown in following picture	
6.1.10	Make sure you have a UART connection between your PC and a mother board, as shown in 6.1.5. Check the “COM” number on you PC.	Go to the Windows Control Panel →Administrative Tools →Computer Management → Device Manager → Ports → USB Serial Port # (connect or disconnect to see the COM port of that module)
6.1.11	Open the “**Tera Term**” serial terminal on you PC.	
6.1.12	Open Tera Term and choose a COM port, which you have found in step 3, and click OK	
6.1.13	Choose Setup->Serial port to configure the Baud rate etc.	

© 2015 Dialog Semiconductor

6.1.14	Set “Baud rate” to 115200, ”Data” to 8 bit, ”Parity” to None, ”Stop” to 1 bit and “Flow control” to none. Click OK. Now we have a properly configured UART terminal on our PC.	
6.1.15	Go back to Keil Project. In the menu bar, select Debug->Start/Stop Debug Session.	
6.1.16	A dialog window pops up, like the one on the right. Please click “OK”	
6.1.17	Press F5 key or click execution button as shown in following picture, to start code execution.	

 © 2015 Dialog Semiconductor

6.1.18	Then you can see a *hello* message on your UART terminal screen. That means you have successfully programmed and started the peripheral program on DA14580/581 Demo board. The *peripheral_setup* demo consists of a small suite of tests that encompasses some of the most commonly used peripherals such as I2C EEPROM, SPI Flash, Rotary Encoder, audio buzzer etc. For more detailed info and technical details please refer to the *UM-B-005: DA14580 Peripheral Examples* as well as the source code of the *peripheral_setup* demo.	

© 2015 Dialog Semiconductor

7 Power Management: measuring current

The design of this DA14580 and DA14581 DEVKT –Basic is made in such a way that the microcontroller can be isolated completely from the rest of the board.

This is illustrated in the block diagram shown in Figure 21. Shown are the connections of the jumpers J4, J5 and J6. For extra info see the electrical schematic in section 0.

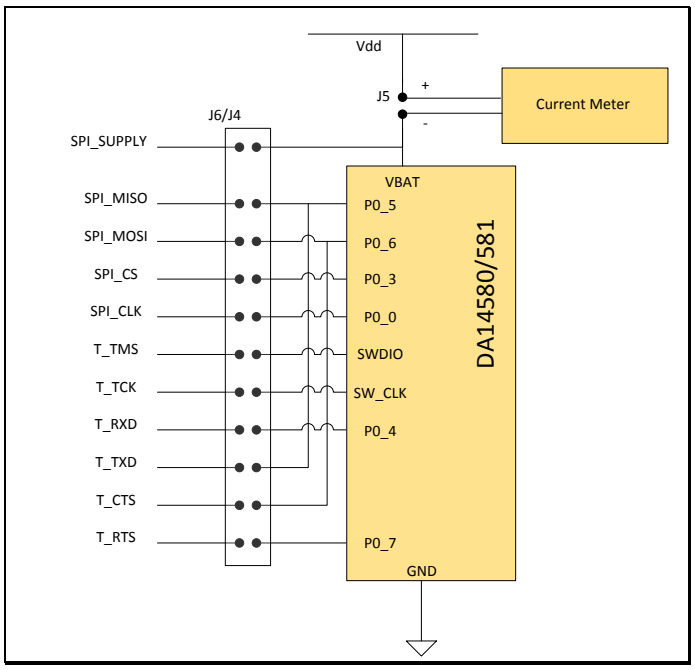


Figure 21: Setup of DA14580/581 DEVKT – BASIC during power measurement

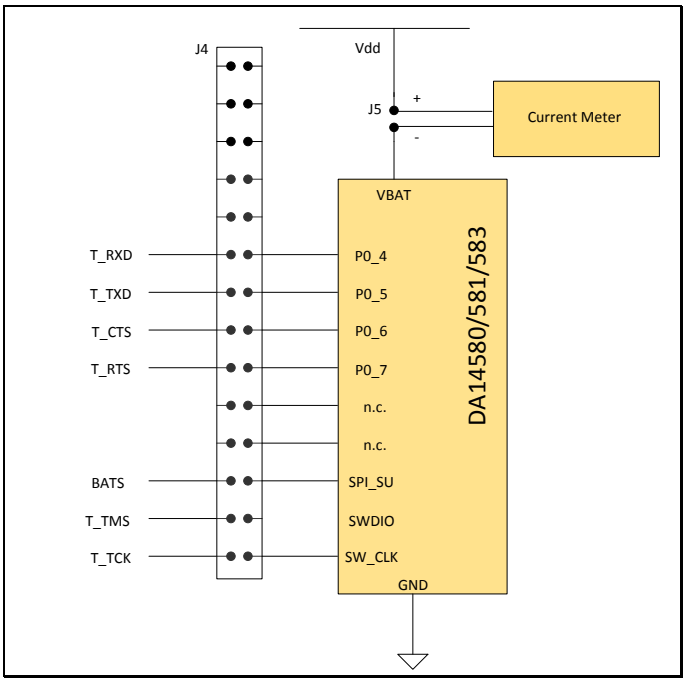


Figure 22: Setup of DA14583 DEVKT – BASIC during power measurement

© 2015 Dialog Semiconductor

Steps how to do the power measurements:

1. Connect the Current Meter to jumper J5.
2. Mount the jumpers needed for downloading the software (see chapter 0 and chapter 5).
3. Download the software.
4. Start the software.
5. Wait till software has reached ‘Deep Sleep’.
6. Dismount **all** the jumpers.
   Now almost all the DA14580/581/583 pins are isolated and only the current meter and GND are connected.
7. Read the current.

For additional info: see AN-B-015 DA14580/581 Supply current measurements. [12]

See chapter 4.4 Deep Sleep current measurement.

Web-link: http://support.dialog-semiconductor.com/system/files/AN-B-015_DA14580_Current_Measurement.pdf

© 2015 Dialog Semiconductor

Appendix A Layout DA14580/581

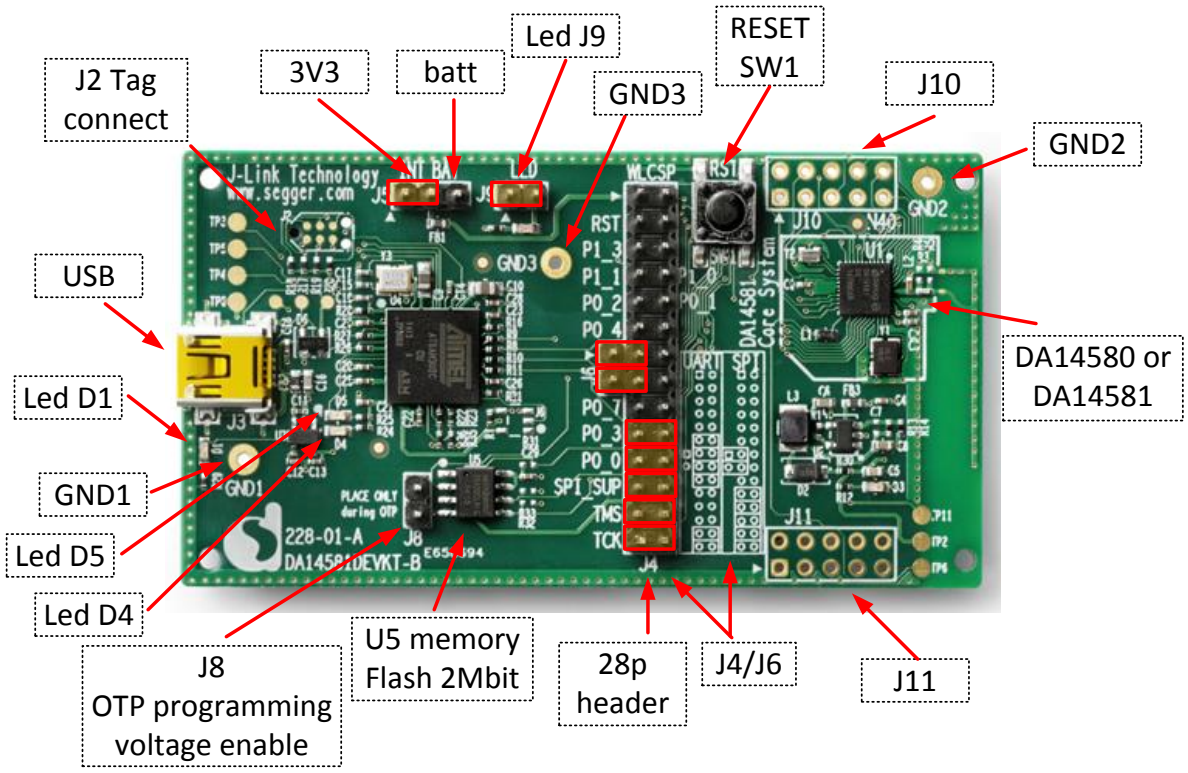


Figure 23: A general overview of the DA14581 development board

Appendix B Connections of J10 and J11

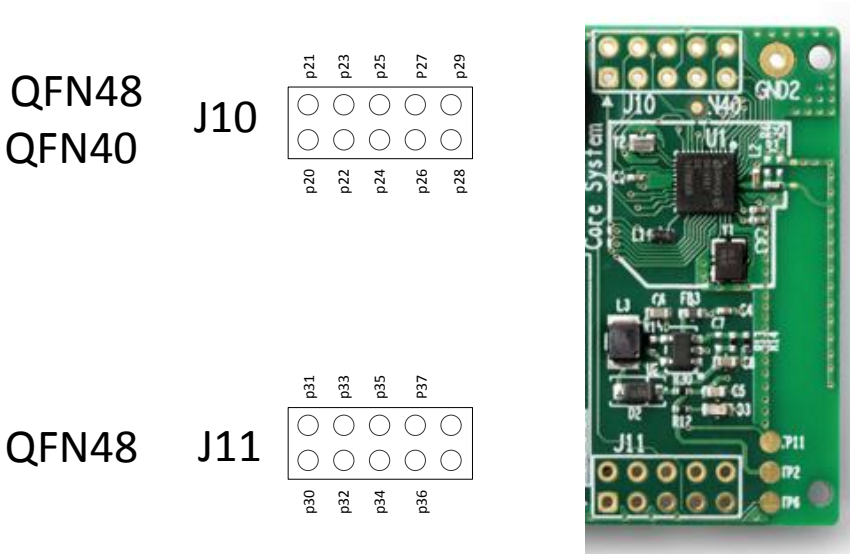


Figure 24: DA14580/581 Connection diagram of J10 and J11

Note 1 The QFN48 package is not available for the DA14581.

© 2015 Dialog Semiconductor

Appendix C Layout DA14583

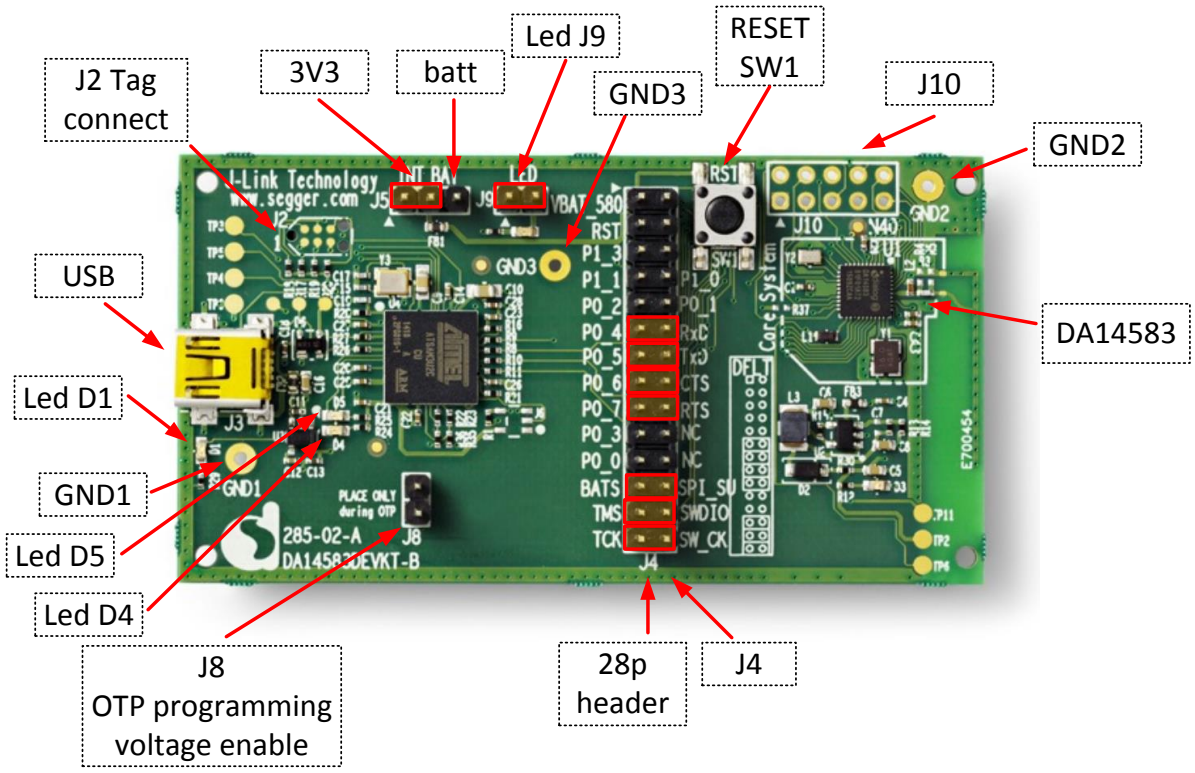


Figure 25: A general overview of the DA14583 development board

Note 1 The QFN48 package is not available for the DA14581.

Appendix D Connections of J10

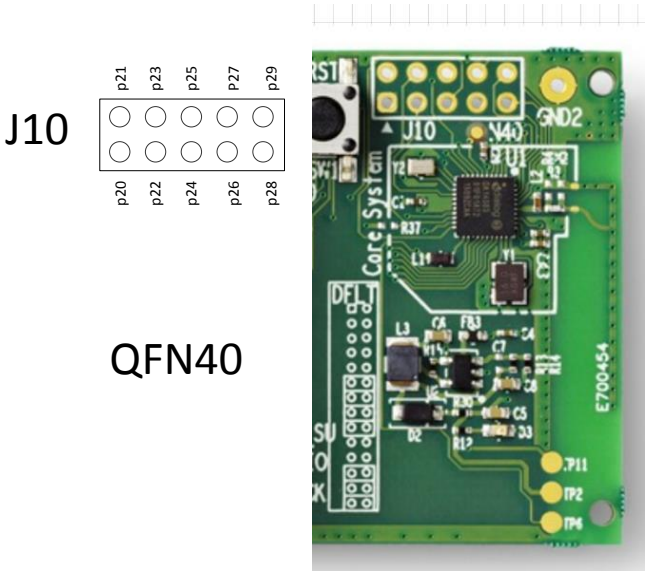


Figure 26: DA14583 Connection diagram of J10

© 2015 Dialog Semiconductor

Appendix E Using the smart snippets CLI

All the information/syntaxes about the CLI can be found under the **HELP** tab in the SmartSnippets GUI or by writing **Smartsnippets –help** in the CLI. In this example, it is assumed that the SPI memory is using P0_0 as SCK, P0_3 as CS, P0_5 as MISO and P0_6 as MOSI. First of all, the CLI can send the commands either via UART or JTAG according to the binary file that has been loaded. On the one hand, if the commands are going to be sent via UART, the following binary file must be used:

- `flash_programmer.bin`

On the other hand, if the commands are going to be sent via JTAG, the following binary file must be used:

- `jtag_programmer.bin`

Note 2 The files can be found in the SmartSnippets resources folder and must be to be downloaded into the DA14580.

Secondly, open the CLI by pushing the Shift button and right click on the **'bin'** folder of the SmartSnippet and select '**Open command window here'**:

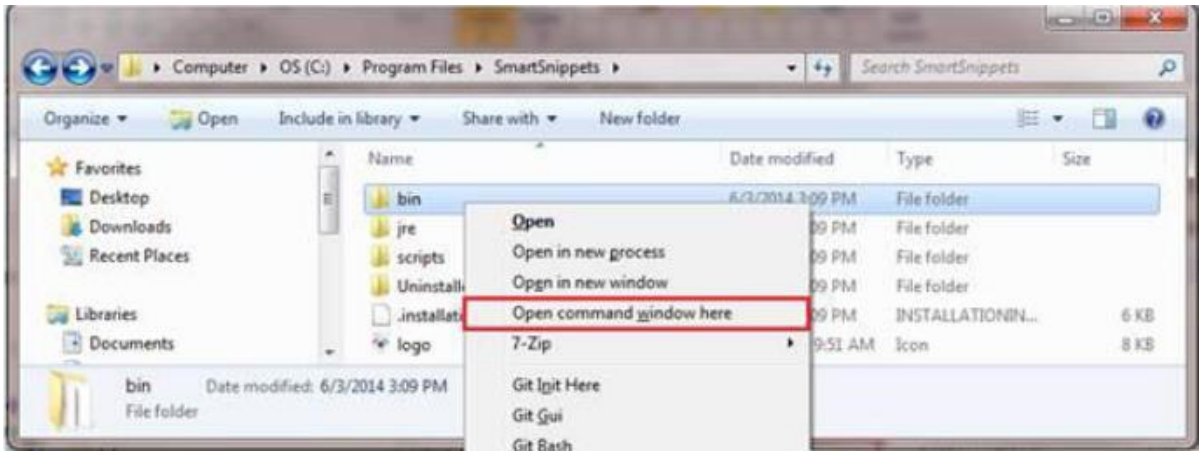


Figure 27: Open the CLI of SmartSnippets

Finally, in order to write a value 0x1347 (example of a Bluetooth device address) at the address 0x93 for instance, the following command line can be written:

SmartSnippets.exe -type spi -chip DA14580-01 -jtag 228202458 -cmd write_field -offset 0x93 - data 1347 -firmware "D:\SmartSnippets\resources\jtag_programmer.bin"

The output should be:

```
Found SWD-DP with ID 0x0BB11477
FPUnit: 4 code (BP) slots and 0 literal slots
Found Cortex-M0 r0p0, Little endian.
BTLE device DA14580 selected.
Using default GPIO pin Id: P1_2.
File c:\users\glagnieu\smartsnippets\ressources\jtag_programmer.bin could not been found.
Using default baudrate: 57600 Bd.
Burned 2 bytes to address 0x00093.
```

Figure 28: SmartSnippets output

© 2015 Dialog Semiconductor

Appendix F Issues with opening a project for the first time

F.1 Issue description:

When a user is working on a Keil uVision project, it can happen that some entries are be missing in the .uvopt file. If this happens, then when the user clicks on the button 'settings' (options{debug tag}) with the{J-LINK/J-TRACE Cortex} selected, uVision crashes.

F.2 Possible causes:

Some important information concerning the j-link driver is missing: calling the driver's dll probably causes the crash. The versions of Keil uVision found to be affected are: versions 5.11.1.0 and 5.10.0.2.

F.3 Under which circumstances user will encounter this error:

When a local GIT repository is first created, this file (.uvopt) does not exist, since it is not included in the remote repository. When the user opens the project for the first time, this file is created, but some keys/values are missing.

F.4 Proposed solution:

Follow the steps below.

1. Ensure that the .uvopt file does not exist in the folder of your project. If it exists and crash has been identified to happen, delete the .uvopt file.
2. Open the Keil project and close it. The .uvopt file is created automatically in the project folder (where the .uvproj is located).
3. Open the .uvopt file, using your favorite text editor.
4. Under the key <TargetOption> add the flowing lines:

```
<TargetDriverDllRegistry>
<SetRegEntry>
<Number>0</Number>
<Key>JL2CM3</Key>
<Name>-U228202424 -O78 -S0 -A0 -C0 -JU1 -JI127.0.0.1 -JP0 -RST0 -N00("ARM
CoreSight SW-DP") -D00(0BB11477) -L00(0) -TO18 -TC10000000 -TP21 -TDS8007 -TDT0 -
TDC1F -TIEFFFFFFFF -TIP8 -TB1 -TFE0 -FO7 -FD20000000 -FC800 -FN0</Name>
</SetRegEntry>
</TargetDriverDllRegistry>
```

5. Save the .uvopt file and close the text editor.
6. Open the Keil project in uVision.
7. Click on *Project→Options for Project 'XXX'.*
8. On the 'Debug' Tab, select J-Link / J-TRACE Cortex debugger and click on the 'Settings' button for the debugger (not the simulator). This is the instance where the crash would happen.
9. The 'Cortex JLink/JTrace Target Driver Setup' Dialog opens*. Select your debugger as normal.*
10. Close the dialog windows clicking ok.
11. *Now, normal operation of j-link debugger is resumed.* After you have finished your work, close the Keil uVision IDE to allow for updates to the *.uvopt file to be saved.*

© 2015 Dialog Semiconductor

Appendix G ‘Blinky’ Test software

```
/**
****************************************************************************************
 *
 * @file main.c
 *
 * @brief Blinky example for DA14580/581 SDK
 *
 * Copyright (C) 2012. Dialog Semiconductor Ltd, unpublished work. This computer
 * program includes Confidential, Proprietary Information and is a Trade Secret of
 * Dialog Semiconductor Ltd.  All use, disclosure, and/or reproduction is prohibited
 * unless authorized in writing. All Rights Reserved.
 *
 * <bluetooth.support@diasemi.com> and contributors.
 * ****************************************************************************************
 */
#include <stdio.h>
#include "global_io.h"
#include "common_uart.h"
#include "periph_setup.h"
#include "gpio.h"

#define LED_OFF_THRESHOLD 10000
#define LED_ON_THRESHOLD  400000

void system_init(void);
void blinky_test(void);
/**
****************************************************************************************
 * @brief  Main routine of the UART example
 *
 ****************************************************************************************
 */
int main (void)
{
    system_init();
    periph_init();
    blinky_test();
    while(1);
}

/**
****************************************************************************************
 * @brief System Initiialization
 *
 *
 ****************************************************************************************
 */
void system_init(void)
{
    SetWord16(CLK_AMBA_REG, 0x00);                // set clocks (hclk and pclk ) 16MHz
    SetWord16(SET_FREEZE_REG,FRZ_WDOG);           // stop watch dog
    SetBits16(SYS_CTRL_REG,PAD_LATCH_EN,1);       // open pads
    SetBits16(SYS_CTRL_REG,DEBUGGER_ENABLE,1);    // open debugger
    SetBits16(PMU_CTRL_REG, PERIPH_SLEEP,0);      // exit peripheral power down
}

/**
 ****************************************************************************************
 * @brief Blinky test fucntion
 *
 *
 ****************************************************************************************
 */

void blinky_test(void)
{
    int i=0;
    // Select function of the port P1.0 to pilot the LED
    printf_string("\n\r\n\r");
    printf_string("***************\n\r");
    printf_string("* BLINKY DEMO *\n\r");
    printf_string("***************\n\r");

    while(1) {
```


© 2015 Dialog Semiconductor

```
        i++;
        if (LED_OFF_THRESHOLD   == i) {
            GPIO_SetActive( LED_PORT, LED_PIN);
            printf_string("\n\r *LED ON* ");
        }
        if (LED_ON_THRESHOLD == i) {
            GPIO_SetInactive(LED_PORT, LED_PIN);
            printf_string("\n\r *LED OFF* ");
        }
                                    if (i== 2*LED_ON_THRESHOLD){
                                            i=0;
                                    }
    }
}
```


© 2015 Dialog Semiconductor

Appendix H Latency Timer of FTDI cable

If an external FTDI cable is used to burn the OTP (or to download the image into the external memory), the Latency Timer of the FTDI cable has to be changed from 15ms to <10ms.
To change the Latency Timer:
Device Manager → COM port → Right click on the COM port chosen → Properties → Port Settings → Advanced → Latency Timer: set it <10ms.

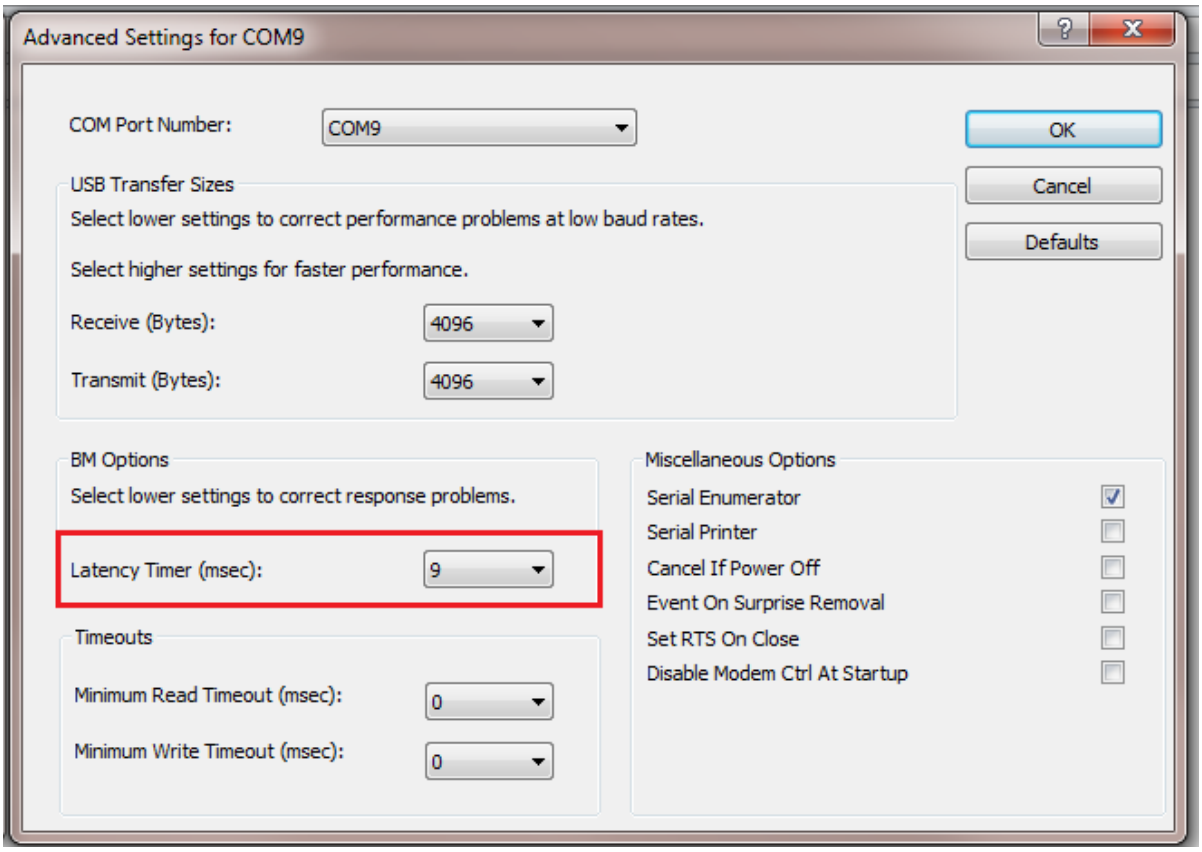


Figure 29: FTDI Latency Timer

© 2015 Dialog Semiconductor

8 Web-Links

- Support Dialog Semiconductors e.g. datasheets and software:
  http://support.dialog-semiconductor.com/resources#tools
- SmartBond DA14580:
  http://www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-da14580
- datasheet DA14580:
  http://support.dialog-semiconductor.com
- SmartBond Reference Designs
  http://www.dialog-semiconductor.com/products/bluetooth-smart/smartbond-reference-designs

9 Revision history

Revision	Date	Description
1.0	30-10-2014	Initial version for DA14580 and DA14581.
2.0	05-11-2014	Merged DA14580 documentation with the DA14581 documentation.
2.1	13-01-2015	Minor text changes and schematic updates.
2.2	01-May-2015	Updated for SDK 3.0.8
2.3.	3-Jun_2015	Updated for SDK 3.0.10 and adding DA14583 drawings

© 2015 Dialog Semiconductor

Status definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

Disclaimer

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including without limitation the specification and the design of the related semiconductor products, software and applications.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Customer notes that nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document are subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, unless otherwise stated.

© Dialog Semiconductor. All rights reserved.

RoHS Compliance

Dialog Semiconductor complies to European Directive 2001/95/EC and from 2 January 2013 onwards to European Directive 2011/65/EU concerning Restriction of Hazardous Substances (RoHS/RoHS2).
Dialog Semiconductor's statement on RoHS can be found on the customer portal https://support.diasemi.com/. RoHS certificates from our suppliers are available on request.

Contacting Dialog Semiconductor

United Kingdom (Headquarters)
Dialog Semiconductor PLC
Phone: +44 1793 757700

Germany
Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:
enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan
Dialog Semiconductor K. K.
Phone: +81 3 5425 4567

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Singapore
Dialog Semiconductor Singapore
Phone: +65 64 849929

China
Dialog Semiconductor China
Phone: +86 21 5178 2561

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8291